z/OS

MVS Diagnosis: Procedures

GA22-7587-00

z/OS

MVS Diagnosis: Procedures

GA22-7587-00

Note

Before using this information and the product it supports, be sure to read the general information under “Appendix. Notices” on page A-1.

First Edition, March 2001

This edition applies to Version 1 Release 1 of z/OS (5694-A01), and to subsequent releases and modifications until otherwise indicated in new editions.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address below.

IBM welcomes your comments. A form for readers’ comments may be provided at the back of this publication, or you may address your comments to the following address:

International Business Machines Corporation
Department 55JA, Mail Station P384
2455 South Road
Poughkeepsie, NY 12601-5400
United States of America

FAX (United States & Canada): 1+845+432-9405
FAX (Other Countries):
Your International Access Code +1+845+432-9405

IBMLink (United States customers only): IBMUSM10(MHVRCFS)
Internet e-mail: mhvrcfs@us.ibm.com
World Wide Web: http://www.ibm.com/servers/eserver/zseries/zos/webqs.html

If you would like a reply, be sure to include your name, address, telephone number, or FAX number.

Make sure to include the following in your comment or note:

- Title and order number of this book
- Page number or topic related to your comment

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

About This Book

This book will help you diagnose problems in the MVS operating system, its subsystems, its components, and in applications running under the system. Using the book, you will be able to:

- Identify the problem type
- Determine the failing subsystem, component, or application
- Collect and analyze data needed to diagnose the problem
- Develop a search argument for searching problem reporting data bases
- Know what problem data is needed before reporting the problem to IBM

This book will help you determine why a problem occurred and where a problem occurred; it does not describe how to fix program instructions in your own code.

Who Should Use This Book

This book is for anyone who diagnoses software problems that occur while running the operating system. This person is usually a system programmer for the installation. This book is also for application programmers who are testing their programs.

The level of detail at which this book is written assumes that the reader:

- Understands basic system concepts and the use of system services
- Codes in Assembler language, and reads Assembler and linkage editor output
- Codes JCL statements for batch jobs and cataloged procedures
- Understands the commonly used diagnostic tasks and aids, such as message logs, dumps, and the interactive problem control system (IPCS)
- Understands the techniques for reporting problems to IBM

How to Use This Book

Use the procedures in this book to diagnose system failures.

If the installation does not wish to debug the problem or does not have the source code involved in the problem, use a diagnostic procedure to collect the problem data needed for reporting the problem to IBM. IBM will debug the problem and provide a fix.

If the installation wishes to debug the problem and has the source code, use a diagnostic procedure to collect problem data. The installation's diagnostician can use this data to debug the problem. If the problem is in IBM code, the diagnostician should report the problem to IBM.

Where to Find More Information

Where necessary, this book references information in other books, using shortened versions of the book title. For complete titles and order numbers of the books for all products that are part of z/OS, see *z/OS Information Roadmap*. The following table lists titles and order numbers for books related to other products.

Short Title Used in This Book	Title	Order Number
A Structured Approach to Describing and Searching Problems	*A Structured Approach to Describing and Searching Problems*	SC34-2129
Introducing the Information/Family	*Introducing the Information/Family*	GC34-4045
Principles of Operation	*Enterprise Systems Architecture/390 Principles of Operation*	SA22-7201
ServiceLink User's Guide	*ServiceLink User's Guide*	SH52-0300
Using Information/Family Problem Management	*Using Information/Family Problem Management*	SC34-4055

Accessing licensed books on the Web

z/OS licensed documentation in PDF format is available on the Internet at the IBM Resource Link Web site at:

```
http://www.ibm.com/servers/resourcelink
```

Licensed books are available only to customers with a z/OS license. Access to these books requires an IBM Resource Link Web userid and password, and a key code. With your z/OS order you received a memo that includes this key code.

To obtain your IBM Resource Link Web userid and password log on to:

```
http://www.ibm.com/servers/resourcelink
```

To register for access to the z/OS licensed books:

1. Log on to Resource Link using your Resource Link userid and password.
2. Click on **User Profiles** located on the left-hand navigation bar.
3. Click on **Access Profile.**
4. Click on **Request Access to Licensed books.**
5. Supply your key code where requested and click on the **Submit** button.

If you supplied the correct key code you will receive confirmation that your request is being processed. After your request is processed you will receive an e-mail confirmation.

Note: You cannot access the z/OS licensed books unless you have registered for access to them and received an e-mail confirmation informing you that your request has been processed.

To access the licensed books:

1. Log on to Resource Link using your Resource Link userid and password.
2. Click on **Library**.
3. Click on **zSeries**.
4. Click on **Software**.
5. Click on **z/OS**.
6. Access the licensed book by selecting the appropriate element.

Using LookAt to look up message explanations

LookAt is an online facility that allows you to look up explanations for z/OS messages and system abends.

Using LookAt to find information is faster than a conventional search because LookAt goes directly to the explanation.

LookAt can be accessed from the Internet or from a TSO command line.

You can use LookAt on the Internet at:

```
http://www.ibm.com/servers/eserver/zseries/zos/bkserv/lookat/lookat.html
```

To use LookAt as a TSO command, LookAt must be installed on your host system. You can obtain the LookAt code for TSO from the LookAt Web site by clicking on **News and Help** or from the *z/OS Collection*, SK3T-4269.

To find a message explanation from a TSO command line, simply enter: **lookat** *message-id* as in the following example:

```
lookat iec192i
```

This results in direct access to the message explanation for message IEC192I.

To find a message explanation from the LookAt Web site, simply enter the message ID. You can select the release if needed.

Note: Some messages have information in more than one book. For example, IEC192I has routing and descriptor codes listed in *z/OS MVS Routing and Descriptor Codes*. For such messages, LookAt prompts you to choose which book to open.

Summary of Changes

Summary of Changes
for GA22-7587-00
z/OS Version 1 Release 1

This book contains information also presented in *OS/390 MVS Diagnosis: Procedures*.

Chapter 1. Introduction

MVS can process large amounts of work efficiently because it keeps track of storage in a way that makes its storage capacity seem greater than it is. It’s a complex system made up of many components, similar to the human body. And, like the human body, MVS can experience problems that need to be diagnosed and corrected.

If an MVS problem occurs, this book can help you determine ***what happened and why it happened***. The following are examples of problems you might encounter while running MVS:

- An abnormal end occurs in processing, known as an abend.
- A job remains hung in the system.
- The system or a process repetitively loops through a series of instructions.
- Output looks nothing like you thought it would.
- Processing slows down.
- Processing stops, requiring that you reIPL.

For system problems, MVS displays symptoms that will help you with your diagnosis. Problem source identification, called PSI, is the determination of what caused the error. Why was an abend issued? What program is using so much of the system storage? What component caused that hang? Which program is looping?

All of these questions can be answered using the procedures documented in this book to diagnose the system problem. There are times, however, when your diagnosis will require the assistance of the IBM Support Center.

Collecting Problem Data

The results of your own problem source identification are useful even if you cannot pinpoint the problem. When you pass a problem on to the IBM Support Center, the information you collect when the problem occurs is very important to help diagnose what went wrong at your installation. Having this information available when you call IBM can save you time because you might not be asked to recreate the problem.

Diagnostic Overview

In many cases, you’ll wait until an error happens two or three times before actually taking the time to diagnose it. You might wait for any number of reasons; just as if you have a short-lived ache or pain you might not go to a doctor. When you decide to diagnose a system problem, follow these steps:

1. When the problem occurs, collect the symptom data and determine what type of problem it is. See “Identifying System Problems” on page 1-2 for more information.
2. Once you determine the type of problem, use the procedures recommended in this book to diagnose the problem. This will help you determine if the problem is a system problem or a user problem. Using these procedures, you can identify the source of the problem.
3. If it is a system problem, build a search argument from the data collected while following the procedure.
4. Perform the search. You might have to refine your search with more data from the problem. See “Searching Problem Reporting Databases” on page 1-4 for more information.

5. If no fix for the problem can be found, report the problem to IBM, using the search argument as a method to determine if the problem is a duplicate. See "Reporting a Problem to IBM" on page 1-13 for more information.

You can approach diagnosis in two ways:

- By problem type
- By type of dump

This book will help you diagnose by problem type. See *z/OS MVS Diagnosis: Tools and Service Aids* for more information about diagnosing by type of dump.

Identifying System Problems

A system problem can be described as any problem on your system that causes work to be stopped or degraded. The steps involved in diagnosing these problems are different for each type of problem.

Before you can begin to diagnose a system problem, however, you have to know ***what kind of problem you have***. To identify a system problem, look at the following:

- System processing witnessed by the operator.
- The dump, in which the system records information about the system problem. It is important to remember that the error triggering a dump might be a symptom itself, and the information needed to diagnose the root cause might not be captured in that dump. Depending on what type of dump the system or the operator takes, you can determine the type of system problem you need to diagnose and whether you will need to collect additional data.
- The logrec data set, which contains a history of the errors encountered by the system.
- The console log.

Table of Problem Indicators

The following table contains examples of indicators of a system problem. Some problems might need to be investigated through more than one diagnostic procedure to find the cause. If there are several indicators, look for the earliest problem that caused the other problems. For example, if there are several abends and a wait state, look for the earliest abend code and begin diagnosis there.

Table 1-1. Indicators of System Problems

Indicator	**System Action**	**System Programmer Action**	**Problem Type**
SVC dump taken	Produces dump	Review dump to determine if further diagnosis is required	ABEND See "Chapter 2. Diagnosing an Abend" on page 2-1.
Message received indicating a system or user abend	• Produces dump • Produces record of error	Review response of system message to determine the abend's impact on the installation.	ABEND See "Chapter 2. Diagnosing an Abend" on page 2-1.

Table 1-1. Indicators of System Problems (continued)

Indicator	System Action	System Programmer Action	Problem Type
Job does not end, no further output is produced, and the job cannot be cancelled.	No response	Use the DUMP command to obtain an SVC dump. If the SVC dump does not provide the necessary information, ask the operator to take a stand-alone dump.	HANG/WAIT or LOOP See either “Chapter 3. Diagnosing a System Hang” on page 3-1 or “Chapter 4. Diagnosing a Loop” on page 4-1.
No response on user’s console	The system might load the following into the PSW: X'070E0000 00000000'	Have the operator check to see if the system console is responsive. If it is not, take a stand-alone dump. If it is, take an SVC dump of the user’s address space.	HANG/WAIT or LOOP See either “Chapter 3. Diagnosing a System Hang” on page 3-1 or “Chapter 4. Diagnosing a Loop” on page 4-1.
No response on operator’s console	The system might load the following into the PSW: X'070E0000 00000000'	Enter a CONTROL Q command to end all multi-line WTOR messages and route all currently queued messages to hardcopy. If the console remains hung, ask the operator to take a stand-alone dump.	HANG/WAIT or LOOP See either “Chapter 3. Diagnosing a System Hang” on page 3-1 or “Chapter 4. Diagnosing a Loop” on page 4-1.
Commands entered from the operator’s console are not received by the system	No response	Ask the operator to take a stand-alone dump	HANG/WAIT or LOOP See either “Chapter 3. Diagnosing a System Hang” on page 3-1 or “Chapter 4. Diagnosing a Loop” on page 4-1.
The system does not write messages on the console with master authority or the alternate console or receive messages from these consoles	No response	Ask the operator to take a stand-alone dump	HANG/WAIT or LOOP See either “Chapter 3. Diagnosing a System Hang” on page 3-1 or “Chapter 4. Diagnosing a Loop” on page 4-1.
Central processor at 100% utilization	Slow processing	Use an online monitor, such as RMF, to determine whether the problem originates from a high priority job in normal processing or from a problem.	LOOP See “Chapter 4. Diagnosing a Loop” on page 4-1.
An ABEND dump is produced	Continue processing	Review the dump to determine if further diagnosis is required.	ABEND (possibly user) See “Chapter 2. Diagnosing an Abend” on page 2-1.
System processing slows.	Processing degrades	Use an online monitor, such as RMF, to determine where the problem originates.	PERFORMANCE DEGRADATION See “Chapter 6. Diagnosing a Performance Problem” on page 6-1.

Table 1-1. Indicators of System Problems (continued)

Indicator	System Action	System Programmer Action	Problem Type
A job is using a high percentage of central processor storage	Processing degrades	Use an online monitor, such as RMF, to determine whether the problem originates from a high priority job in normal processing or from a problem.	LOOP See “Chapter 4. Diagnosing a Loop” on page 4-1.
Output is missing or is incorrect.	Processing continues	Determine if input is correct.	OUTPUT PROBLEM See “Chapter 5. Diagnosing an Output Problem” on page 5-1.
A job remains in the system for a long time and does not end	A bottleneck occurs	If the job is not a low priority job in a busy system, cancel the job and submit it again. If the error recurs, use RMF to determine where the bottleneck is occurring. Use the DUMP command to obtain an SVC dump.	HANG/WAIT See “Chapter 3. Diagnosing a System Hang” on page 3-1.
The operator, on the system console, witnesses a series of WAIT indicators followed by a burst of activity	Processing continues	Use an online monitor, such as RMF, to determine where the bottleneck is occurring.	ENABLED WAIT or PERFORMANCE DEGRADATION See either “Chapter 3. Diagnosing a System Hang” on page 3-1 or “Chapter 6. Diagnosing a Performance Problem” on page 6-1.
A message appears on the system console that indicates a wait	No response	Ask the operator to take a stand-alone dump	HANG/WAIT See “Chapter 3. Diagnosing a System Hang” on page 3-1.

Searching Problem Reporting Databases

While you are diagnosing a system problem, you will collect data about that problem. What was the abend code? What did the registers and PSW contain at the time of error? What is the failing module or CSECT? In what component was the problem found?

The answers to these questions are the material for a *search argument*. A search argument is a list of *symptoms* for a problem. A search argument is also called a *symptom string*.

Search arguments are used to search problem reporting databases. If the problem being diagnosed was already reported and the symptoms entered into the database, the search will produce a match. Each procedure in this book contains a step titled Known Problem in which you are requested to perform a search. If the search finds a match, you can request the problem fix, if available. Diagnosis stops at this point.

To perform a search, do the following:

1. If your installation has a problem reporting database, search it first.
2. If not or if the search provided no useful information, search an IBM problem reporting database, if one is available. See “Formats for Symptoms” for more information.
3. Otherwise, give a RETAIN search argument to the IBM Support Center. IBM will do the search.

IBM searches the RETAIN database, which contains the search arguments from problem incident reports and from problem solutions in authorized program analysis reports (APAR). ServiceLink and Info/Management enable searches of the RETAIN database.

References

- See *ServiceLink User’s Guide* for more information about ServiceLink.
- See *A Structured Approach to Describing and Searching Problems* for search arguments
- See *Introducing the Information/Family* for programs used in problem searches.
- See *Using Information/Family Problem Management* for Info/Management panels and search symptoms

The following topics are contained in this section:

- “Formats for Symptoms”, which distinguishes between the types of symptom formats
- “Developing Search Arguments” on page 1-6, which describes how to develop a search argument while you are performing diagnosis
- “Search Argument Symptoms” on page 1-7, which lists the symptoms used in search arguments

Formats for Symptoms

The symptoms composing search arguments used by IBM for searching problem reporting databases are in two formats. The table in this chapter shows both formats.

- **Free-format symptom**: Use these symptoms with a free-format search tool, such as ServiceLink. Free-format symptoms are usually used when searching the RETAIN database.

  An example of a module CSECT name as a free-format symptom is: `IEAABCD`
- **RETAIN symptom**: Use RETAIN symptoms:
  - With a tool such as Info/Management to search the RETAIN database
  - When reporting a problem to IBM
  - In descriptions of problems in APARs and program temporary fixes (PTF)

  RETAIN symptoms are also called *structured symptoms* and *failure keywords*. An example of a module CSECT name as a RETAIN symptom is: `RIDS/IEAABCD`

A third format, the **MVS symptom**, is used by dump analysis and elimination (DAE) when determining if a dump is a duplicate of a previous dump; MVS symptoms are not used for searching problem databases. These symptoms are contained in the DAE data set. An example of a module CSECT name as an MVS symptom is: `CSECT/IEAABCD`

RETAIN or MVS symptoms consist of:

- A prefix keyword: 1 to 8 characters used to identify the type of symptom

- Data: At least 1 character

All characters are alphanumeric, #, @, or $. The maximum length, including the prefix and slash, for a RETAIN symptom is 15 characters and for an MVS symptom is 50 characters.

References

- See “Search Argument Symptoms” on page 1-7 for a table of free-format and RETAIN symptoms
- See *z/OS MVS Diagnosis: Reference* for the keywords for RETAIN and MVS symptoms

Developing Search Arguments

Obtaining Arguments

Obtain search arguments from an SVC dump, SYSMDUMP ABEND dump, or stand-alone dump by using IPCS subcommands.

Subcommand	Dump Output Heading	Symptom Format
STATUS FAILDATA	Search Argument Abstract	RETAIN
VERBEXIT DAEDATA	DUMP ANALYSIS AND ELIMINATION (DAE)	RETAIN, MVS
VERBEXIT LOGDATA	SEARCH ARGUMENT ABSTRACT	RETAIN
VERBEXIT SYMPTOM	Primary Symptom String	RETAIN

RETAIN symptoms are also available from logrec software records by using EREP to produce a detail edit report. The symptoms are under the heading SEARCH ARGUMENT ABSTRACT.

Based on the RETAIN and MVS symptoms, you can create free-format symptoms, if you plan to use them.

References

- See *z/OS MVS IPCS Commands* for the subcommands
- See *z/OS MVS Diagnosis: Reference* for logrec record formats
- See *EREP User’s Guide* for EREP formatting of logrec records

Number of Symptoms in an Argument

For most problems, use three to five symptoms in the search argument. If the first search produces no matches, remove some symptoms and search again. If the first search produces too many matches, add one or more symptoms and search again. Also, try different symptoms. Searching is an iterative process.

Selecting Symptoms

The following are suggestions for selecting symptoms:

- When selecting search symptoms, use only repeatable data. For example, a particular abend will always produce the same abend code. In contrast, the address in a program status word (PSW) can be different each time the problem recurs.
- Symptoms about data areas are useful for identifying a problem. Use the names of a data area and the incorrect field in the data area as symptoms.
- If searching does not produce a match, remove some symptoms or use different symptoms and try again.

Order of Symptoms in an Argument

The symptoms in each diagnostic procedure in the following chapters are listed in the order that provides the most efficient search.

Search Argument Symptoms

The following table lists, in alphabetical order, the symptoms used in search arguments. The table lists free-format and RETAIN symptoms for both software and hardware problems. The hardware symptoms are included for completeness. You can use these symptoms to search problem reporting databases for previous occurrences of errors.

For the symptoms for a particular problem, see the procedure for that problem; the symptoms in the procedures are listed in the order that provides the fastest search.

Symptom Data

In the symptom formats, the type of variable data is indicated by the lowercase letter, as follows. All uppercase letters must be used exactly as shown.

b	Bit
c	Alphanumeric character
d	Decimal number
h	Hexadecimal number

The length of the data, if fixed, is indicated by the number of characters or, if variable, is specified in the format. For fixed-length data, pad with zeros on the left, if needed.

Problem Data	Free-Format Symptom	RETAIN Symptom
Abend code: • 3-digit hexadecimal system completion code for an abnormal end • 4-digit decimal user completion code from an abnormal end	ABENDhhh ABENDdddd For example: ABEND78A ABEND1234	AB/S0hhh AB/Udddd For example: AB/S078A AB/U1234
Address: • Address in storage • Virtual address • Offset in a module, field, or data area		ADRS/hhhhhhhh For example: ADRS/00053186
APAR (IBM authorized program analysis report) number		PZFS/ccddddd For example: PZFS/PP12345
Bill of materials (BOM) for a device		BOMS/ddddddd For example: BOMS/1234567
Bits per second transmission rate		BPS/ccccccccccc (1 to 11 alphanumeric characters) For example: BPS/9600
Device card/frame/board location		CLS/ccccccccccc (1 to 11 alphanumeric characters) For example: CLS/1200034

Problem Data	Free-Format Symptom	RETAIN Symptom
Device type	D/Tdddd For example: D/T3390	DEVS/ccc (generic device) DEVS/ddddccc (specific device) For example: DEVS/PRT DEVS/3800003
ECA (device engineer change announcement)		ECAS/ccc For example: ECAS/021
FDS (fix distribution media) including the upgrade tape number or program temporary fix (PTF) PUT tape number		FDS/cccccccdddd FDS/ccccUdddd For example: FDS/5740CB10201 FDS/5752U8109
FEA (device optional or special feature)		FEAS/dddd For example: FEAS/0034
Field name: • Data field • Control block • Field in a data area • Label in a DSECT • Footprint in a system diagnostic work area (SDWA) variable recording area (VRA)	cccccccccc For example: TCB SDWAVRA 3E	FLDS/cccccccccc (1 to 10 alphanumeric characters) For example: FLDS/TCB FLDS/TCBCMP FLDS/SDWAVRA VALU/H3E
Halt code issued by the system or a device		HL/Scccccccccc (1 to 10 alphanumeric characters) HL/Dhhhh HL/Ehhhh For example: HL/SDASH7
I/O operation code, channel command word (CCW) flag, or teleprocessing operation code		IOPS/cccccccc (command code or CCW flag - 1 to 8 alphanumeric characters) IOPS/hh (teleprocessing operation code) For example: IOPS/010201 IOPS/80
Level of the system, component, or failing module, including: • Base system release, that is, the version, release, and modification numbers • Program product level • Function modification identifier (FMID) • Engineering change level (ECL) for a device	cccdddd ARddd Rddd For example: HBB5520 AR520 R520	LVLS/cccdddd LVLS/dddddd For example: LVLS/HBB5520 LVLS/000520 LVLS/520
Major unit code identifier for a device		MAUS/dddd For example: MAUS/1801
Map entry point location		MEP/cccccccc (1 to 8 alphanumeric characters) For example: MEP/A0010

Problem Data	Free-Format Symptom	RETAIN Symptom
Message identifier or text General messages, such as IEA995I, are not useful as search symptoms.	MSGcccccccccccc For example: MSGIEF187I	MSG/ccccccccccc (1 to 11 alphanumeric characters) For example: MSG/IEF187I MSG/NOID (for a message without an identifier)
Minor unit code identifier for a device		MIUS/dddd For example: MIUS/7070
Operation code of a failing instruction in assembler language mnemonic or hexadecimal machine language	cccccccc hh For example: BCR 07	OPCS/cccccccc (1 to 8 alphanumeric characters) OPCS/hh For example: OPCS/BCR OPCS/07
Overlaid storage, data area, or field	OVERLAY cccccccccc For example: OVERLAY TCB	OVS/cccccccccc (1 to 10 alphanumeric characters) For example: OVS/TCB
Identifier of: • Footprint • Work area • Header • Data set • Device • Failing job • Failing step • Failing command • JCL statement • Feature • Product	cccccccccc For example: DISPLAY	PCSS/cccccccccc (1 to 10 alphanumeric characters) For example: PCSS/DISPLAY
Identifier of: • The product and component. The first 4 characters identify the product; the last 5 characters identify the component. • The program that caused the problem. • A system service used by the failing component.	ccccccccc For example: 5655SC1C5	PIDS/ccccccccc For example: PIDS/5655SC1C5

Problem Data	Free-Format Symptom	RETAIN Symptom
Name of: • CSECT that failed • Load module that failed • Analytic procedure • Program to run • Program that called a failing module Do not use the RIDS symptom for an entry point name. If a search with the CSECT name produces no matches, omit the name. Some problems, such as damaged data areas, can cause failures in many modules. Use the CSECT name, if available, in preference to the load module name. Load module names are less likely to produce matches. Often an SVC dump contains one or two RIDS symptoms. However, if the failing component does not supply the failing CSECT name or module name, the system determines the name, if possible, and places it in the RIDS symptom. If the name cannot be determined, the system places the load module name IEANUC0x in the RIDS symptom.	cccccccc For example: IEFVTGLU	RIDS/cccccccc (1 to 8 alphanumeric characters) For example: RIDS/IEFVTGLU
Part name or number, if the part is missing, extra, or defective		PNS/ccccccccccc (1 to 11 alphanumeric characters) PNS/ddddddd For example: PNS/BOARD PNS/0123456
PTF (program temporary fix) Use the PTF if the module was recently updated.	ccccccc For example: UY00934	PTFS/ccccccc For example: PTFS/UY00934
Publication identifier, including a document order number or the identity of an area in a document, for example, a chapter and a page	ccdddddd ccddddddv CHAPTERd PAGEdd For example: GC281839 GC2818391 CHAPTER9 PAGE5	PUBS/cccccccccc (1 to 10 alphanumeric characters) For example: PUBS/GC281839 PUBS/GC2818391 PUBS/CHAP9 PUBS/PAGE5

Problem Data	Free-Format Symptom	RETAIN Symptom
Reason or return code generated by a program, including a step code, status code, feedback code, or condition code Reason and return codes are 4 bytes long, but may appear in books and messages in other lengths.	RChhhhhhhh or RChh RSNrrrrrrrr or RSNrr For example: RC0000002A RC20 RSN00000040 RSN08	PRCS/hhhhhhhh For example: PRCS/0000002A
REA (request engineering action) number		REAS/ccccccc For example: REAS/00123BB
Record format		RECFM/ccc (1 to 3 characters) For example: RECFM/VBA
Record organization		RECO/ccc For example: RECO/SEQ RECO/DIR
Record type, including hardware control records, record labels, and data records	cccccccccc For example: SMF TYPE 41	RECS/cccccccccc (1 to 10 alphanumeric characters) For example: RECS/SMF41
Register associated with a problem or the register/PSW difference	GRdd ddhhh For example: GR02 0000AD34 090A4	REGS/ddhhh where dd = decimal register number where hhh = offset from PSW address at time of error REGS/ccdd where cc = AR: access register CR: control register FP: floating point register GR: general purpose register For example: VALU/H0000AD34 REGS/090A4
RPQ (request for price quotation) number		RPGS/ccccccc For example: RPGS/00143BB
Signal to warn the operator issued by the system or a device		SIG/ccccccccccc (1 to 11 alphanumeric characters) For example: SIG/MACHINE SIG/CHECK
Status indication for a device		STA/ccccccccccc (1 to 11 alphanumeric characters) For example: STA/UNIT STA/CHECK

Problem Data	Free-Format Symptom	RETAIN Symptom
Value in a field, register, or message text	cccccccc For example: 03658723	VALU/Bbbbbbbbb (1 to 8 bits) VALU/Ccccccccc (1 to 8 characters) VALU/Hhhhhhhhh (1 to 8 hexadecimal digits) VALU/hhhhhhhh (Register content - 8 hexadecimal digits) For example: VALU/B101010 VALU/CMLPA VALU/H01F6 VALU/03658723
Wait state code issued by the system or a device	WAIThhh For example: WAIT03C	WS/D0hhh (disabled wait or coded wait state) WS/E0hhh (enabled wait) For example: WS/D003C WS/E0008

Example: ServiceLink Search

The example below shows a search argument of ABEND0C4 IEAVEDS0 used on ServiceLink. Note that ServiceLink uses free-format symptoms.

```
SISMM1                   Service Information Search

Type an APAR/PTF number or search information, then press Enter.

APAR/PTF number . . . . . . . . . . . _______
Search argument
ABEND0C4 IEAVEDS0___________________________________
Updated on or after . . . . . . . . . mm/dd/yy
  Last entry  . . . . . . . . . . . :
  Last time used  . . . . . . . . . :
Closed on or after. . . . . . . . . . mm/dd/yy
  Last entry  . . . . . . . . . . . :
  Last time used  . . . . . . . . . :
Sort? . . . . . . . . . . . . . . . . 1 1. Yes
                                        2. No
Restrict search?  . . . . . . . . . . 1 1. APARs only
                                        2. PTFs only
                                        3. No
Program . . . . . . . . . . . . . . . _______ +
Change library selections?  . . . . . 2 1. Yes
                                        2. No
```

Example: ServiceLink Matches Found

The following output shows the matches found based on the ServiceLink searches listed above.

```
SISLT4                   Items in Service Libraries
                                                   Items 1  to
Type an action code next to one or more items, then press Enter.
  B=Open for browse  R=Order  T=Track  P=Output
                                                          More:
    ---------------------- MVS and associated products ----------------
            Closing
Act Item    Code    Abstract
 _  OY63816  PER    CSDSYSND MAY BE LEFT ON IF A MEMTERM OCCURS WHILE A
 _  II07091         ABEND0C4 RC11 IN IEAVEDS0 ON MVCL FROM FRR EXTENSIO
 _  OY61888  PER    WAITA00 ON D/T3090 S MACHINE RUNNING UNDER PR/SM LP
 _  II07061         ABEND066 RC28 IECVPST CUSP ICUCDR01
 _  II03292  CAN    THIS APAR CONTAINS INFORMATION ON RUNNING NETVIEW R
 _  OY42953  SUG    ABEND0C4 WAIT01C ABEND0C6 ABEND0C1 TASK TERMINATION
 _  II06870         ABEND0C4 PGM11 IEAVEDS0
 _  OY56421  MCH    3090-S MACHINE HAS WAITA00 RUNNING UNDER PR/SM LPAR
 _  OY27728  PER    IEAVSETS LOOPS AFTER ABEND0C4 IN IEAVEDS0
 _  OY50251  PER    OVERLAY OF TCB CAUSES ABEND0C4 IN IEAVEDS0 WHEN USI
 _  OY19065  PER    DUMMY WAIT SVC DUMP RECOVERY LEAVES SYSTEM NONDISPA
 _  OY26492  PER    IEAVSETS OBTACT ROUTINE ON BOTH CP0 AND CP2
 _  OY25668  PER    IEAVEDS0 ABEND0C4 OVERLAY ASCBLSMQ BY CSVVFTCH 89/0
 _  OY41650  PER    IEAVEDS0 UY90699 SRBFLNK POINTER BAD ABEND0C4
 _  OY37666  PER    LOOP IN IEAVRTI0 (IEA0TI00) WHILE ENQUEUING A TQE T
 _  OY05910  SUG    ASCBFAIL IS LEFT ON AFTER AN ABEND0C4 IN IEAVEDS0 E
 _  II04505  USE    ABEND0C4 IN IEAVEDS0 WHEN ACCESSING AN SRB POINTED
 _  OY12020  SUG    ABEND0C4 IN IEAVEDS0 ATTEMPTING TO REFERENCE TCB P
 _  OY18644  SUG    WAIT084 WAIT09X RECOVERY LOOP DUE TO PSAMODEW OVERL
 _  OY03987  PER    LOOP IN IGC079 ON ASCBCPUS AFTER PROGRAM CHECK DUE
```

Reporting a Problem to IBM

IBM recommends that you report a problem to IBM when your search does not find a match in your installation’s databases. The IBM Support Center will search an IBM database to find a fix for the problem. If the problem is new, the IBM Support Center will assist in the diagnosis.

This section lists problem data recommended for calling IBM. This section organizes the problem data by problem type.

Preparing Dump Data

Before calling IBM, format the dump using all the IPCS subcommands recommended in the appropriate procedure. Some of the subcommands take time to run for a large dump. Also, start the IPCS session before calling. Then, during the call, the output can be browsed as needed.

Collecting Data

If possible, obtain all data about a problem soon after the problem. Otherwise, updates to the system can cause discrepancies in the data.

Automatic Problem Reporting

Reporting a Problem

Parts of the system automatically report the need for service to IBM; for example, the central processor complex (CPC) reports problems directly to IBM. If the system contains a Hardware Management Console, you should be aware that problems in the Sysplex Timer and in direct access storage devices (DASDs) may be automatically reported, even though the problems are recorded by MVS:

- MVS captures information about the problems and creates logrec records:
  - ETR record: For problems in the Sysplex Timer
  - DASD-SIM record: For problems in DASD
- For a unique Sysplex Timer or DASD error, the Hardware Management Console creates a problem record (PMR) in RETAIN to notify IBM that service is needed.

Name and Level of the Operating System(s)

To report most problems, you should provide to the IBM Support Center the name and level of the operating system(s). Perform the following analysis procedure to determine the systems being used:

1. **Determine the level of the MVS system**. Use the IPCS subcommand CBFORMAT with the CVT control block to determine the product level.

Example: CBFORMAT CVT Output

In the following output, the PRODN field indicates an MVS operating system level of SP 3.1.3 and the PRODI field indicates the FMID as JBB3313.

```
CVT: 00FC5C40
   -0028  PRODN.... SP3.1.3   PRODI.... JBB3313   VERID....
   -0006  MDL...... 3090      RELNO.... 038
   +0000  TCBP..... 00000218  0EF00.... 00FE6148  LINK..... 00FD9A64
   +000C  AUSCB.... 00FC6220  BUF...... 00000000  XAPG..... 00FEFE10
   +0018  0VL00.... 00FFBAAE  PCNVT.... 00FEBE80  PRLTV.... 00FEBDEC
```

   Reference

   See *z/OS MVS IPCS Commands* for more information about the CBFORMAT subcommand.

2. **Determine if the system is running as a uniprocessor or multiprocessor**. In IPCS STATUS WORKSHEET output, just above the heading `PROCESSOR RELATED DATA`, find:

   ```
   No. of active CPUs: 000n
   ```

   The *n* indicates the number of processors running.

Example: STATUS WORKSHEET Output

In the following output, there are three active processors.

```
CSD  Available CPU mask: E000  Alive CPU mask: E000  No. of active CPUs: 0003
```

3. **Determine if the system is running as a VM guest**. Near the top of the STATUS WORKSHEET output, find:

   ```
   CPU Model nnnn Version hh Serial no. nnnnnn Address nn
   ```

An MVS system can run directly on the processor or under PR/SM or as a VM guest. The version in STATUS WORKSHEET depends on where the dump is requested:

- A dump in the MVS system running as a VM guest has `Version FF`.
- A dump in the MVS system running directly has some other value as the version.

  The central processor identifier is identified in PCCACPID. To see PCCACPID, use the IPCS CBFORMAT PCCA(x) subcommand, where x is a processor. The version immediately follows the control block eye-catcher of PCCA.

Example: STATUS WORKSHEET Output

In the following output, the version is 32, indicating that the system is not running as a VM guest.

```
                             MVS Diagnostic Worksheet

Dump Title: SLIP DUMP ID=0001


CPU Model 3090 Version 32 Serial no. 281114 Address 02
Date: 03/29/1993   Time: 11:09:30  Local

Original dump dataset: SYS1.DUMP00
```

4. **For a system running as a VM guest, determine the level of the VM product, by using VMSES/E.**

   Reference

   See *z/OS SMP/E User’s Guide* for using SMP.

5. **Determine if the system is running under PR/SM**. Format the dump with the following IPCS subcommand:

   ```
   CBFORMAT CVT STRUCTURE(CVT)
   ```

   Obtain the address of the service call control block (SCCB) from the CVTSCPIN field. Then format the dump with the following IPCS subcommand to obtain the SCCB:

   ```
   LIST address LENGTH(4096)
   ```

   Look at the SCCBBFY bit, which is the first bit in the SCCBCON1 field in the SCCB. If this bit is on and the system is not a VM guest, then the MVS system is running under Processor Resource/Systems Manager (PR/SM).

   Reference

   See *z/OS MVS Data Areas, Vol 4 (RD-SRRA)* for the format of the SCCB.

Problem Data Examples

When you report a problem to IBM, provide as much information as possible. The response time for having a problem solved is much quicker when problem data is reported with the call to IBM. The following tables show examples of the type of data than can be reported for each type of system problem.

Problem Data for an Abend

Problem data	Example
Problem type	Abend
Search argument	PIDS/5752SC1CH RIDS/IGVGLSQA#L RIDS/IGVGLSQA AB/S010A PRCS/20 REGS/090AF RIDS/IGVGLERR#R PCSS/GETMAIN
Dump, formatted by IPCS, online or printed	
Failing input request: macro, command, or statement	GETMAIN macro
SDWAVRA keys, lengths, and contents	Key: VRAEBC Length 29 Contents: UNABLE TO OBTAIN DYNAMIC AREA
Offset of the failing instruction into the module	0AF
Accompanying messages: identifiers and texts	IEA705I ERROR DURING GETMAIN, SYS CODE=10A-20 MYJOB23 STEP2 00 000708AB 00891234 00894321 02352000 00001000
SYS1.LOGREC report, if used	
All printed output and output data sets related to the problem	
Data on any related problems	
Module name and level	IGVGLSQA at PTF level UY00934
Name and level of the operating system(s) with a list of program temporary fixes (PTF) applied at the time of the problem and all installation modifications, exits, and products with other than Class A service	Level HBB4420 (MVS/ESA System Product 4.2) at put tape level T8804
Other problem data developed while using the diagnosis book for the component, subsystem, or program	

Problem Data for a Documentation Problem

Problem data	Example
Problem type	Documentation
Search argument	PIDS/5752SC1CH PUBS/GC281822 PCSS/GETMAIN PCSS/SP PUBS/GC2818220 PUBS/APPLICATION PUBS/DEVELOPMENT PUBS/MACRO PUBS/REFERENCE PUBS/PAGE125
Document title	MVS/ESA Application Development Reference: Services for Assembler Language Programs
Order number for the book, supplement, or technical newsletter	GC28-1644-0
Page or topic containing the problem	Page 125 or Topic 3.2.1.4
Version and release level for the system, subsystem, product, or program installed in the system and documented	Installed: MVS/ESA System Product 4.2 Documented: MVS/ESA System Product 4.2

Problem Data for a Hang or Wait

Problem data	Example
Problem type	Disabled wait
Search argument	PIDS/5752SC1CW RIDS/ILRTMI00 WS/D003E
Dump, formatted by IPCS, online or printed	
SMF records, if obtained	
Accompanying messages: identifiers and texts	ILR073W AUXILIARY STORAGE BACKING SLOT SHORTAGE, ASM UNABLE TO CONTINUE
Hard-copy log, beginning 15 to 30 minutes before the problem, or master trace, if not wrapped between the problem and dump	
All printed output and output data sets related to the problem	
Module name and level	ILRTMI00 at PTF level UY00934
Name and level of the operating system(s) with a list of program temporary fixes (PTF) applied at the time of the problem and all installation modifications, exits, and products with other than Class A service	Level HBB4420 (MVS/ESA System Product 4.2) at put tape level T8804
Other problem data developed while using the diagnosis book for the component, subsystem, or program	

Problem Data for a Loop

Problem data	Example
Problem type	Disabled loop or enabled loop
Search argument	PIDS/5752SC1CW RIDS/ILRTMI00#L RIDS/ILRRMI00
Dump, formatted by IPCS, online or printed	
Range of the loop	Address X'FB0816' through X'FB0932'
Accompanying messages: identifiers and texts	ILR007I DUPLEX PAGE DATA SET BAD, DUPLEXING SUSPENDED
Hard-copy log, beginning 15 to 30 minutes before the problem, or master trace, if not wrapped between the problem and dump	
All printed output and output data sets related to the problem	
Module name and level	ILRTMI00 at PTF level UY00934
Name and level of the operating system(s) with a list of program temporary fixes (PTF) applied at the time of the problem and all installation modifications, exits, and products with other than Class A service	Level HBB4420 (MVS/ESA System Product 4.2) at put tape level T8804

Problem Data for a Message

Problem data	Example
Problem type	Message
Accompanying messages: identifiers and texts	

Reporting a Problem

Problem data	Example
Search argument	PIDS/5752SC1CH MSG/IEA705I VALU/CODE=10A-29 RIDS/IGVRV SM PCSS/GETMAIN
Failing input request: macro, command, or statement	GETMAIN macro
All printed output and output data sets related to the problem	
Module name and level	IGVRVSM at PTF level UY00934
Name and level of the operating system(s) with a list of program temporary fixes (PTF) applied at the time of the problem and all installation modifications, exits, and products with other than Class A service	Level HBB4420 (MVS/ESA System Product 4.2) at put tape level T8804
Other problem data developed while using the diagnosis book for the component, subsystem, or program	

Problem Data for an Output Problem

Problem data	Example
Problem type	INCORROUT, that is, incorrect, incomplete, or missing output in a data set, message, return code, or reason code
Search argument	PIDS/5752SCI32 PCSS/COPYDUMP PCSS/OUT023 MSG/BLS18169I VALU/COPIED
All input associated with the problem, including all data sets, commands, macros, and statements	
All output associated with the problem, including data sets, reports, and records	
JCL for all data sets involved	
Source code for any exit routine involved	
Accompanying messages: identifiers and texts	BLS18169I DUMP 123 IS BEING COPIED
Hard-copy log, beginning 15 to 30 minutes before the problem, or master trace, if not wrapped between the problem and dump	
SYS1.LOGREC records, beginning 15 to 30 minutes before the problem and edited using the SPOTCHK and TIMESEQ parameters	
Name and level of the operating system(s) with a list of program temporary fixes (PTF) applied at the time of the problem and all installation modifications, exits, and products with other than Class A service	Level HBB4420 (MVS/ESA System Product 4.2) at put tape level T8804
Other problem data developed while using the diagnosis book for the component, subsystem, or program	

Problem Data for a Performance Problem

Problem data	Example
Problem type	Performance
Search argument	PIDS/5752SC1CH PCSS/GETMAIN
Dump, formatted by IPCS, online or printed	

Problem data	Example
System responses to DISPLAY and JES commands	
The IEAIPSxx, IEAICSxx, and IEAOPTxx members, if analyzed	
SMF data and data from an online monitor, such as RMF	
All printed output and output data sets related to the problem	
Name and level of the operating system(s) with a list of program temporary fixes (PTF) applied at the time of the problem and all installation modifications, exits, and products with other than Class A service	Level HBB4420 (MVS/ESA System Product 4.2) at put tape level T8804
Other problem data developed while using the diagnosis book for the component, subsystem, or program	

Problem Data for a Return Code

Problem data	Example
Problem type	Return code
Search argument	PIDS/5752SC1CH PCSS/GETMAIN PRCS/20 PRCS/08 RIDS/IGVLSQA#L RIDS/IGVGLSQA
Dump, formatted by IPCS, online or printed, if available	
Failing request: macro, command, or statement or program invocation	GETMAIN macro
Accompanying messages: identifiers and texts	IEA705I ERROR DURING GETMAIN, SYS CODE=10A-20 MYJOB23 STEP2 00 000708AB 00891234 00894321 02352000 00001000
Data on any related problems	
All printed output and output data sets related to the problem	
Module name and level	IGVGLSQA at PTF level UY00934
Name and level of the operating system(s) with a list of program temporary fixes (PTF) applied at the time of the problem and all installation modifications, exits, and products with other than Class A service	Level HBB4420 (MVS/ESA System Product 4.2) at put tape level T8804
Other problem data developed while using the diagnosis book for the component, subsystem, or program	

Chapter 2. Diagnosing an Abend

Symptoms

- Messages have indicated a system or user abend. For example, message IEA995I has been issued to the operator console.
- A dump was produced.
- An error was recorded in the logrec data set.

What You’ll Need

- IPCS installed
- SVC dump, SYSUDUMP, SYSMDUMP, or SYSABEND dump
- Logrec error record
- Master trace
- Job log

About Using the Procedure

Use this procedure to diagnose these two *coded wait states:*

- Wait state X'040'
- Wait state X'064' with X'09' in the second byte of the right half of the program status word

Use the procedure in “Chapter 4. Diagnosing a Loop” on page 4-1 to diagnose the following abends:
X'071'
X'322'

Use the procedure in “Chapter 3. Diagnosing a System Hang” on page 3-1 to diagnose the following abends, though they could also be loops:
X'122'
X'222'

Use this procedure to diagnose all other abends.

Abend Analysis

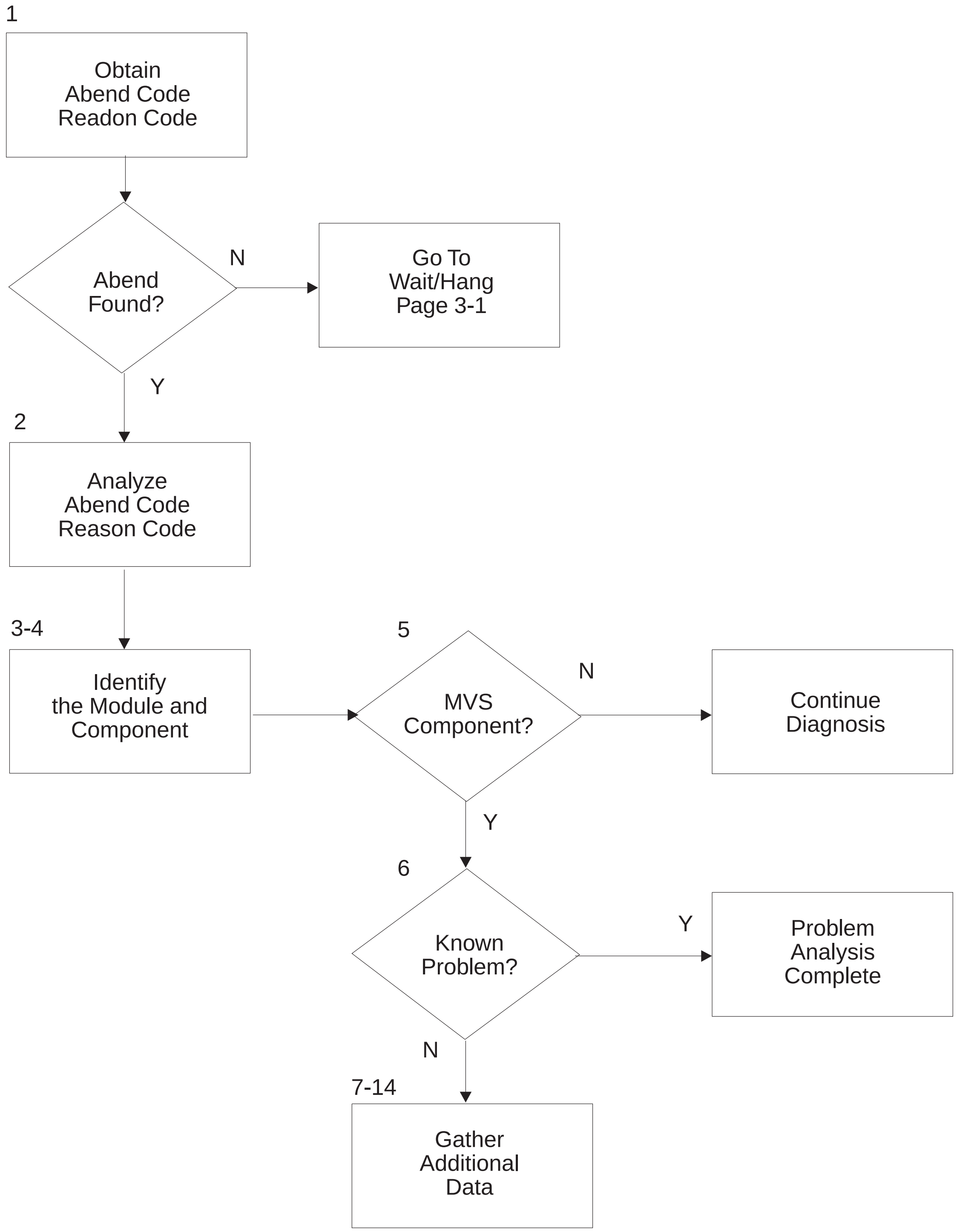

Figure 2-1. Diagnosing an Abend

Step 1: Obtain Abend Code and Reason Code

Obtain the abend code and a reason code, if provided.

1A. If you already have the abend and reason code, go to step “Step 2: Analyze Abend Code and Reason Code” on page 2-5.

1B. If you do not have the abend and reason code in a message, look for the abend and reason code in one of the following:

- ***STATUS FAILDATA output from a dump:*** For the abend code, look for AB/S0hhh in the symptoms under the heading SEARCH ARGUMENT ABSTRACT, if present.

  For the reason code, look for PRCS/hhhhhhhh in the symptoms, or look in the register indicated in the abend code explanation.

  Example: STATUS FAILDATA Output

  In the following output, the symptom string indicates a system abend code of X'03C' with a reason code of X'2D000810'.

  ```
    PIDS/5752SC1CR RIDS/NUCLEUS#L RIDS/IARYTASS AB/S003C PRCS/2D000810
    REGS/0E0B REGS/088FA RIDS/IARRR#R

    SYMPTOM            DESCRIPTION
    -------            -----------
    PIDS/5752SC1CR     PROGRAM ID: 5752SC1CR
    RIDS/NUCLEUS#L     LOAD MODULE NAME: NUCLEUS
    RIDS/IARYTASS      CSECT NAME: IARYTASS
    AB/S003C           SYSTEM ABEND CODE: 003C
    PRCS/2D000810      ABEND REASON CODE: 2D000810
    REGS/0E0B4         REGISTER/PSW DIFFERENCE FOR R0E: 0B4
    REGS/088FA         REGISTER/PSW DIFFERENCE FOR R08: 8FA
    RIDS/IARRR#R       RECOVERY ROUTINE CSECT NAME: IARRR

  OTHER SERVICEABILITY INFORMATION

    RECOVERY ROUTINE LABEL:  IARRRCV
    DATE ASSEMBLED:          10/06/92
    MODULE LEVEL:             HBB4430
    SUBFUNCTION:             REAL STORAGE MANAGEMENT
  ```

  Reference: See *z/OS MVS IPCS Commands* for information about the STATUS FAILDATA subcommand.

- ***Logrec error records:*** Obtain the SOFTWARE RECORD for the abend:
  – From the logrec data set, by formatting the records in the data set with EREP to obtain detail edit reports
  – From the dump for the abend, by entering the VERBEXIT LOGDATA subcommand

  To find the SOFTWARE RECORD for the problem:
  – Look for the record with the most recent time stamp.
  – Match its symptoms and other data to the dump title or the STATUS FAILDATA output.
  – Match the ERRORID in the record to the Error ID in the STATUS WORKSHEET dump output. The ERRORID consists of a sequence number, the central processor identifier, the ASID, and the time stamp. The same ERRORID in several records means that the records are for the same problem. Look in the first of these records.

  For the abend code, look for AB/S0hhh in the symptoms under the heading SEARCH ARGUMENT ABSTRACT, if present.

For the reason code, look for PRCS/hhhhhhhh in the symptoms, or look in the register indicated in the abend code explanation.

Reference: See *z/OS MVS Diagnosis: Tools and Service Aids* for information about the logrec data set.

- ***VERBEXIT SYMPTOM output from a dump:*** Format the dump completely, as described in “Step 9: Gather Additional Data, Recreate” on page 2-9. Look for AB/S0hhh and PRCS/hhhhhhhh in the symptoms.

Example: VERBEXIT SYMPTOM Otuput

In the following output, the primary symptom string indicates a system abend code of X'03C' and a return code of X'2D000810'.

```
  Primary Symptom String:

    RIDS/NUCLEUS#L RIDS/IARYTASS PIDS/5752SC1CR AB/S003C RIDS/IARRR#R
    VALU/HC0099680 REGS/0E0B4 REGS/088FA PRCS/2D000810 VALU/CNAGEMENT

    Symptom            Symptom data       Explanation
    ---------------    -------------      -----------
    RIDS/NUCLEUS#L     NUCLEUS#L          Routine identifier
    RIDS/IARYTASS      IARYTASS           Routine identifier
    PIDS/5752SC1CR     5752SC1CR          Component identifier
    AB/S003C           003C               ABEND code - system
    RIDS/IARRR#R       IARRR#R            Routine identifier
    VALU/HC0099680     C0099680           Error related hexadecimal value
    REGS/0E0B4         0E0B4              Program register
    REGS/088FA         088FA              Program register
    PRCS/2D000810      2D000810           Return code
    VALU/CNAGEMENT     NAGEMENT           Error related character value

The dump does not contain a secondary symptom string.
```

- ***Dump title***: Look at the dump title; some titles contain the abend and reason codes. Obtain the title as follows:
  - For an SVC dump that is still in the SYS1.DUMPxx data set in which it was written, ask the operator to enter a DISPLAY DUMP,TITLE,ERRDATA command to obtain summary information. Look in the reply to the operator for the abend and reason codes and the registers. An SVC dump may also remain captured in dataspace storage. To view information about a dump captured in storage, enter the DISPLAY DUMP command with an ERRORID.
  - For a dump that has been copied from the SYS1.DUMPxx data set, use a STATUS WORKSHEET subcommand to obtain the dump title.

Example: Portion of STATUS WORKSHEET Output

In the following output, the dump title indicates a system abend code of X'08D'.

```
                          MVS Diagnostic Worksheet

Dump Title: ABEND=08D,COMPON=CONVERTER,COMPID=SC1B9,ISSUER=IEFNB9CR

CPU Model 3090 Version FF Serial no. 373577 Address 03
Date: 02/08/1993   Time: 10:26:38  Local

Original dump dataset: SYS1.DUMP03
```

  - For a dump you are viewing in IPCS, use the LIST TITLE subcommand to obtain the dump title.
- ***Job log or VERBEXIT MTRACE output from a dump:*** Look for message IEA995I or other messages with an abend code in the message text. The message might also give a reason code.

In message texts, an abend code can be called a *system completion code* or a *SYS CODE*. A message can show an abend code in the variable text without identifying it as an abend code; use the message explanation to understand the variable text.

References

See the following for system messages:
- *z/OS MVS System Messages, Vol 1 (ABA-AOM)*
- *z/OS MVS System Messages, Vol 2 (ARC-ASA)*
- *z/OS MVS System Messages, Vol 3 (ASB-BPX)*
- *z/OS MVS System Messages, Vol 4 (CBD-DMO)*
- *z/OS MVS System Messages, Vol 5 (EDG-GFS)*
- *z/OS MVS System Messages, Vol 6 (GOS-IEA)*
- *z/OS MVS System Messages, Vol 7 (IEB-IEE)*
- *z/OS MVS System Messages, Vol 8 (IEF-IGD)*
- *z/OS MVS System Messages, Vol 9 (IGF-IWM)*
- *z/OS MVS System Messages, Vol 10 (IXC-IZP)*

If none of these locations contains an abend code, use the procedure in "Chapter 3. Diagnosing a System Hang" on page 3-1.

Step 2: Analyze Abend Code and Reason Code

Look at the explanation of the abend and reason code.

If you cannot fix the problem with the explanation in the *z/OS MVS System Codes* book, or if you cannot tell whether the problem was caused by an IBM or an installation problem, continue with the next step.

Reference: See *z/OS MVS System Codes* for an explanation of the abend and reason code.

Step 3: Identify the Module

Identify the module that abended and the offset of the failing instruction into the module.

Do one of the following, depending on the problem data available.

- ***For an SVC dump, obtain the component name, component identifier, and module name from the dump title.***
- ***Obtain the PIDS/ccccccccc and RIDS/cccccccc symptoms from the search argument.*** PIDS is the program identifier, which is the 4-character product identifier and the 5-character component identifier. RIDS identifies the module.
- ***If the search argument in an SVC dump does not identify the program and module or if the problem involves multiple abends, analyze the dump for the failing task or service request.***

  Reference: See *z/OS MVS Diagnosis: Tools and Service Aids* for information about analyzing an SVC dump.
- ***Obtain the address in the right half of the program status word (PSW) in STATUS FAILDATA dump output.*** The leftmost bit of the leftmost digit denotes addressing mode and is not part of the address.

  Subtract the instruction length from the PSW address to obtain the address of the failing instruction. Do not subtract the instruction length in the following cases; the failing instruction is at the PSW address.

  - Program check interruptions for which the processing of the instruction identified by the old PSW is nullified. See *Principles of Operation* for the interruption action. Some examples are:
    - Page translation exception.
    - Segment translation exception.
    - Vector operation interruption.
    - Access register translation exception.
  - ***Obtain the PSW and registers for the error from the STATUS CPU REGISTERS subcommand.***

Example: STATUS CPU REGISTERS Output

In the following output, the address can be found in the second half of the PSW.

```
CPU STATUS:

 PSW=440C4000  81151A18  (RUNNING IN AR, KEY 0,  AMODE 31, DAT ON)
     DISABLED FOR I/O EXT
   ASID(X'0001') 01151A18. IEANUC01.IARYT+2918 IN READ ONLY NUCLEUS
  ASCB1 at FD0D00, JOB(*MASTER*), for the home ASID
  ASXB1 at FD0FB0 for the home ASID. No block is dispatched
  HOME ASID: 0001 PRIMARY ASID: 0001 SECONDARY ASID: 0001

  GPR VALUES
      0-3  00000000  00000000  0000004E  0141D9B8
      4-7  01311000  01311000  00000000  012DBEE0
      8-11 0115111E  0115011F  8114F120  00000000
     12-15 0000108A  01B535F0  81151964  00000000
```

STATUS CPU REGISTERS supplies the name of the failing module and its offset without having to do a separate WHERE subcommand.

Reference: See *z/OS MVS IPCS Commands* for information about IPCS subcommands.

Do one of the following:

- If analyzing the dump interactively, use the instruction address in a WHERE subcommand to obtain the name of the load module and the offset of the address into the load module. If the module name is not proceeded with IEANUC01, then IPCS has given the load module name. If you enter the STATUS CPU REGISTERS subcommand, a WHERE is performed automatically.

  Use the AMBLIST service aid to list the CSECTs in the load module. Use the offset into the load module to identify the CSECT containing the failing instruction. Then subtract the starting address of the CSECT from the instruction address to obtain the offset into the CSECT.
- If analyzing printed output, find the instruction address:
  - In dump output from the LIST or VERBEXIT SUMDUMP subcommand. Look for the CSECT name eye-catcher. IBM module names are generally followed by an assembly date and a product identifier or PTF level, such as HBB3310 or UY01234; most eye-catchers are at the beginning of a module, but some are at the end.
  - In a module listed for the LPAMAP or VERBEXIT NUCMAP subcommand. LPAMAP will list load modules. Use AMBLIST to obtain the offsets of CSECTS within those load modules. NUCMAP lists CSECTs with offsets, but can only be used for modules within the nucleus.

    Subtract the starting address of the CSECT from the instruction address to obtain the offset into the CSECT.

Step 4: Identify the Component

From the module prefix, identify the component, subsystem, or product, if provided by IBM.

Example: Module Prefix Identification

Using the information in the following output from *z/OS MVS Diagnosis: Reference*, you can determine what component was involved in an error from the module prefix. For instance, an ADY module prefix indicates a DAE-type error.

```
Module       Component                          Product   Component
Prefix         Name                               ID         ID
 ADF      TSO and TSO/E session manager         5665       28505
 ADY     Dump analysis and elimination (DAE)    5752       SC143
```

Reference: See *z/OS MVS Diagnosis: Reference* to find the IBM component, subsystem, or product for a program identifier or module prefix.

Book for the IBM subsystem, product, or command: The product name is in the edition notice inside the front cover.

Step 5: MVS Component?

Continue diagnosis as follows

- For a component of MVS, continue with “Step 6: Known Problem?”.
- For an IBM subsystem or product, continue diagnosis with the subsystem or product.
- For an installation-provided program, including an installation exit routine, continue diagnosis with that program, using the dump for the abend.

Reference: See *z/OS MVS Diagnosis: Tools and Service Aids* for dump analysis of a problem in an installation-provided program.

Step 6: Known Problem?

Search a problem reporting data base to determine if the problem was previously reported.

Use the argument under SEARCH ARGUMENT ABSTRACT in the:

- Logrec detail edit report
- STATUS FAILDATA output
- VERBEXIT LOGDATA output

Example: Search Argument Abstract

Use the information from the search argument abstract as a symptom string to determine if the problem already exists.

```
SEARCH ARGUMENT ABSTRACT

  PIDS/5752SC1CR RIDS/NUCLEUS#L RIDS/IARYTASS AB/S003C PRCS/2D000810 REGS/0E0B
  REGS/088FA RIDS/IARRR#R
```

If an argument is not provided, use the following symptoms:

- Program identifier: PIDS/ccccccccc
- CSECT name or module name: RIDS/cccccccc
- System abend code or user abend code: AB/S0hhh or AB/Udddd
- Abend reason code: PRCS/hhhhhhhh
- Input request (macro, command, or statement): PCSS/cccccccccc

Note: You can also use free-format search arguments. For more information, see “Searching Problem Reporting Databases” on page 1-4.

If the search finds no match, remove some symptoms or add some symptoms. Search again. Continue searching for matches by adding or removing symptoms.

If the search finds that the problem was previously reported, request the problem fix. If not, continue with the next step. Use problem data from following steps to create more symptoms; use these symptoms in later searches.

Reference: See “Searching Problem Reporting Databases” on page 1-4 for more information.

Step 7: Gather Additional Data, Messages and Logrec

Collect and analyze messages and logrec records about the problem.

Use the ERRORID from the dump message and time stamps to select messages and software, symptom, and hardware logrec records related to the problem. Look in the following:

- The job log
- A TSO/E user’s ISPF transaction log or session manager log
- The hardcopy log, also known as the system log (SYSLOG)
- VERBEXIT MTRACE dump output, which shows the buffer for system messages
- VERBEXIT LOGDATA dump output, which formats the logrec buffer
- The logrec data set, formatted by EREP

Look for the following

- Symptom dump message IEA995I for a previous, related abend
- Messages identifying a failing program with a nonzero return code
- I/O error messages

References

- See *z/OS MVS IPCS Commands* for the VERBEXIT MTRACE subcommand.
- See *EREP User’s Guide* for formatting logrec records.
- See *z/OS MVS System Messages, Vol 6 (GOS-IEA)* for message IEA995I.
- See *z/OS MVS System Messages, Vol 9 (IGF-IWM)* for the IOS messages.

Step 8: Gather Additional Data, Abending Program

Identify the program being processed when the abend occurred.

- Obtain the job name from the following:
  SUMMARY output
  VERBEXIT LOGDATA output
  Messages in the job log

Messages in VERBEXIT MTRACE output
SELECT output

Example: SELECT Output

In the following output, the job name NVAST in address space 0073 contains an error.

```
ASID JOBNAME  ASCBADDR  SELECTION CRITERIA
---- -------- --------  ------------------
0073 NVAST    00F6B600  CURRENT ERROR
```

- If a batch job was being processed, obtain the program name from the PGM parameter on the JCL EXEC statement.

Example: PGM Parameter of JCL Statement

In the following output, the PGM parameter indicates that the program name is UNIBANK.

```
//BANK1 EXEC PGM=UNIBANK,PARM='@PLANID=1,10S,SHR',
//             REGION=1024K,COND=(8,LE)
//BANKLOG  DD  DSN=NULLFILE,DISP=SHR
```

- If interactive work was being processed, use the command being processed to identify the program.

References

- See *z/OS MVS IPCS Commands* for the subcommands: for information about the SELECT, SUMMARY, VERBEXIT LOGDATA, and VERBEXIT MTRACE subcommands.
- See *z/OS MVS JCL Reference* for the PGM parameter.
- See *z/OS TSO/E Command Reference* for interactive commands.

Step 9: Gather Additional Data, Recreate

Collect dump data.

If a dump was not written for the abend, re-create the problem and obtain a dump by doing one of the following:

- Set a SLIP command to obtain an SVC dump.

  Reference: See *z/OS MVS System Commands* for information about setting a SLIP trap.
- Insert a DD statement in the JCL for the abending job step to obtain one of the following types of ABEND dumps:
  - SYSMDUMP
  - SYSUDUMP
  - SYSABEND

  Reference: See *z/OS MVS JCL Reference* for information about the SYSMDUMP, SYSUDUMP, or SYSABEND DD statement.

Use IPCS to look at the dump. Use IPCS subcommands in the order indicated by the following list. If using IPCS interactively for an SVC dump, respond yes to the IPCS message that asks if summary data can be used by dump access.

1. STATUS FAILDATA
2. STATUS SYSTEM

Example: STATUS SYSTEM Output

In the following output, IEAVTSDT indicates that this SVC dump was scheduled. Also note the date and time the dump was taken.

```
SYSTEM STATUS:
  Nucleus member name: IEANUC01
  I/O configuration data:
    IODF data set name: IODFST.IODFBF
    IODF configuration ID: MVS90J
    EDT ID: 00
  Sysplex name: XCFLOCAL
  TIME OF DAY CLOCK: A7C9D005 A0675C03  07/15/93 22:18:44 GMT
  TIME OF DAY CLOCK: A7C99A60 A0675C03  07/15/93 18:18:44 local
  Program Producing Dump: SVCDUMP
  Program Requesting Dump: IEAVTSDT
```

3. STATUS CPU REGISTERS DATA CONTENTION
4. STATUS WORKSHEET
5. SUMMARY FORMAT
6. VERBEXIT LOGDATA
7. VERBEXIT SUMDUMP
8. SYSTRACE
9. VERBEXIT MTRACE

Example: VERBEXIT MTRACE Output

In the following output, message IEF450I indicates a system abend of X'522' with a reason code of X'00'.

```
0001 007A5F54  N 0000000 ANO3     93039 10:26:08.31          00000281
               IEA989I SLIP TRAP ID=X13E MATCHED
0001 007A5F54  N 0000000 ANO3     93039 10:26:08.34          00000281
               IEA989I SLIP TRAP ID=X13E MATCHED
0001 007A5F54  N 0000000 ANO3     93039 10:26:08.43          00000281
               IEA989I SLIP TRAP ID=X13E MATCHED
0001 007A5F54  N 0000000 ANO3     93039 10:26:08.49          00000281
               IEA989I SLIP TRAP ID=X13E MATCHED
0001 007A5F54  N 4000000 ANO3     93039 10:26:09.21 TSU05807 00000091
               IEF450I LASSEC2 AAIRACF AAIRACF - ABEND=S522 U0000
               REASON=00000000
0001 007A7430  N 4000000 ANO3     93039 10:26:09.45 TSU06038 00000091
               IEF450I LAMMLF AAIUSER AAIUSER - ABEND=S522 U0000
               REASON=00000000
0001 007A7430  M 4000000 ANO3     93039 10:26:09.59 TSU05807 00000090
               IEF377I LASSEC2 AAIRACF AAIRACF
0001 007A5F54  E                                         064 00000090
               LASSEC2.SPFLOG1.LIST NOT CATLGD 2
0001 007EC02C  N 4000000 ANO3     93039 10:26:09.66 TSU05807 00000090
               /HASP395 LASSEC2 ENDED
0001 007A79C0  N 0200000 ANO3     93039 10:26:10.06 TSU05807 00000081
               /HASP250 LASSEC2 IS PURGED
```

10. Subcommands selected from the list below
11. VERBEXIT SYMPTOM

Before the VERBEXIT SYMPTOM subcommand, add other IPCS subcommands, depending on the problem indicated in the abend explanation or accompanying messages. Pick the subcommands from the following list.

Problem Involves	**IPCS Subcommand**
Allocation/unallocation of jobs	`VERBEXIT ALCMWAIT`
Asynchronous operations manager (AOM)	`VERBEXIT AOMDATA 'TRCDUMP'`
Auxiliary storage	`ASMCHECK`

	`VERBEXIT ASMDATA`
Availability management	`VERBEXIT AVMDATA`

Callable service requests	`CBFORMAT addr STRUCTURE(CSRCPOOL)`
	`CBSTAT addr STRUCTURE(CSRCPOOL)`

Communications	`COMCHECK`

Cross-system coupling facility (XCF)	`COUPLE SUMMARY ALL`
	`COUPLE DETAIL ALL`
	`COUPLE EXCEPTION ALL`

Data-in-virtual	`DIVDATA SUMMARY ALL`
Data lookaside facility of VLF	`DLFDATA`
Global resource serialization	`VERBEXIT GRSTRACE`
Input/output	`IOSCHECK`
JES2	`VERBEXIT JES2`
JES3	`VERBEXIT JES3`
MVS message service (MMS)	`VERBEXIT MMSDATA`

z/OS UNIX System Services (OMVS)	`OMVSDATA SUMMARY`
	`OMVSDATA DETAIL`
	`OMVSDATA EXCEPTION`

Real storage	`RSMDATA SUMMARY`
	`RSMDATA EXCEPTION`

Storage management sub-system (SMS)	`VERBEXIT SMSDATA`
System resources manager (SRM)	`VERBEXIT SRMDATA`
Time sharing option (TSO)	`VERBEXIT TSODATA`
Virtual storage	`VERBEXIT VSMDATA`
Virtual lookaside facility (VLF)	`VLFDATA`

Example: RSMDATA SUMMARY Output

Use the RSMDATA SUMMARY output to get a summary of real image usage in the system. Use the RSMDATA EXCEPTION report to determine where errors might have occurred.

```
Warnings regarding STRUCTURE(PVT) at ASID(X'0001') 00FE0300:
   Storage not in dump
ASID(X'0001') 00FE0314 not available for PVT
RSM processing on a non-stand-alone dump may generate inconsistent data and
false validity check failures.
Data space information may be incomplete for RSM.  Storage not in dump.
One or more major RSM data areas received a validity check warning.  Run
RSMDATA EXCEPTION report.

                     R S M   S U M M A R Y   R E P O R T

                          Tot real Below Prf real Dbl real      Expanded
                          -------- ----- -------- -------- -------------
In configuration . . . .    28,672 4,096   28,640        -        61,440
Available for allocation    27,568 4,078   28,542      148        61,440
Allocated  . . . . . . .    27,558 4,031   28,575      143        60,973
Percent usage  . . . . .        99    98       99       96            99
Common fixed frames  . .     2,737   323    2,737        -             -
  Percent of available .         9     7        9        -             -
Total fixed frames . . .    10,716 1,379        -        -             -
  Percent of available .        38    33        -        -             -

V=R Region:
  First frame number X'00005'
  Last frame number X'00024'
  Size (in frames)              32

Total disabled reference (DREF) pages in real and expanded:           2,488

System currently in "available frame queue low (AVQLOW)" condition.
```

Step 10: Gather Additional Data, Failing Processor

Check for a failing processor in a multiprocessing system.

Look in the VERBEXIT SYMPTOM output for FLDS/CSDACR. This symptom means that a processor was failing and alternate CPU recovery was active. In this case, an abend X'0F3' is generated when alternate CPU recovery abends the program running on the failing processor.

Step 11: Gather Additional Data, Variable Recording Area

Collect the VRA data in the SDWA.

In the STATUS FAILDATA output, VERBEXIT LOGDATA output, or logrec detail edit report, find the heading VARIABLE RECORDING AREA (SDWAVRA). Obtain the SDWAVRA data. For some components, the data consists of a key, a length, and the contents.

Example: Variable Recording Area from STATUS FAILDATA

In the following output, the VRA key is X'1A' and the length is X'94'.

```
VARIABLE RECORDING AREA (SDWAVRA)

  +000    Key: 1A     Length: 94
  +002    02000000    08004000    00040001    1D00E610     |...... .......W.|
  +012    E0001300    00000000    00000000    00000000     |\...............|
  +022    00000000    00000000    00000000    00000000     |................|
  +032    00000000    00000000    00000000    0000002B     |................|
  +042    00100000    00000000    00000000    00000000     |................|
  +052    00000000    00000000    00000000    FFFC0000     |................|
  +062    012DBD64    00000000    01A9CD24    00000C80     |.........z......|
  +072    00000000    019D6690    012DC698    00000000     |..........Fq....|
  +082    01B42000    00000000    00000000    00000000     |................|
  +092    00000000                                         |....            |
```

STATUS FAILDATA will not format an SDWA for a dump requested by SLIP. If SDWA data is not in the dump, obtain problem data from STATUS CPU REGISTERS or view the SDUMP 4K SQA buffer.

References

- See *z/OS MVS Data Areas, Vol 5 (SSAG-XTLST)* for VRAMAP, which describes the VRA keys.
- See *z/OS MVS System Commands* for information about the SLIP command.

Step 12: Gather Additional Data, System Trace Table

Analyze the system trace table, which is formatted by the SYSTRACE subcommand.

A system trace provides a record of system events. Use it to create a picture of the processing occurring at the time of the error.

Starting at the end of the trace, back up to the entry for the abend being diagnosed:

- *SVC D or *SVCE D in the IDENT CD/D columns
- The abend code in the right 3 bytes in the UNIQUE-3 column
- The reason code in the UNIQUE-1 column

Example: SYSTRACE Output

In the following output, the *SVC D indicates that an abend code has been loaded for processor 02. When examining system trace output, look for RCVY entries which represent entry into a recovery routine following an error or interruption.

```
PR ASID TCB-ADDR  IDENT CD/D PSW----- ADDRESS-...
                  .
                  .
                  .
02 000D 006F8E88   SSRV  12D          8120BFD8  006F8E88 000B0000 00000000
                                                00000000
01 002E 006BEE88   SVC    30 070C3000 827FAF36  00000000 00000001 072CFBF4
01 002E 006BEE88   PC    ...   0      81157326  00100
01 002E 006BEE88   PT    ...   0      811B782C  002E
02 000D 006F8E88   DSP       070C0000 8101A9B0  00000000 0001035C 000295A8
02 000D 006F8E88  *SVC     D 070C0000 8101A9B2  80D12090 0001035C 000295A8
02 000D 006F8E88   PC    ...   0      811D7626  00506
01 002E 006BEE88   SVCR   30 070C3000 827FAF36  00000000 00000028 0080002E
01 002E 006BEE88   SVC    38 070C3000 827FAF64  00000000 00000028 072CFBF4
01 002E 006BEE88   SVC     A 070C1000 80F8146A  00000000 FD000236 80F81468
```

In the entry, note the processor in the PR column, the address space identifier in the ASID column, and the task control block (TCB) address in the TCB-ADDR column. The ASID should be the same as the ASID identified in STATUS FAILDATA or STATUS CPU output.

Continue backing up, looking for the following entries:

- The entry for the system service being processed when the abend occurred, if the abend occurred during system processing. This entry will have SVC, SSRV, or SVCE in the IDENT column and the same ASID as the abend entry.
- Problem entries, which have an asterisk (*) before the identifier in the IDENT column.
- Other entries for the same processor, PR column.
- Other entries for the same address space, ASID column.
- Repeated requests by a program for one or more system services. This pattern indicates a loop. An enabled loop has multiple types of entries before the pattern repeats. Continue diagnosis with the program requesting the system services.

References

- See *z/OS MVS IPCS Commands* for the subcommands: for the STATUS and SYSTRACE subcommands.
- See *z/OS MVS Diagnosis: Tools and Service Aids* for the formats of system trace entries.

Step 13: Gather Additional Data, Multiple ERRORIDs

Analyze the problem data for multiple problems.

Collect data for related problem ERRORIDs that occur in a similar timeframe. You can find this data in the logrec data set. The time stamps are a few seconds before or after the time stamp for the abend being diagnosed. The data involves the following:

- The same job step
- The same task (TCB) or service request (SRB)
- The same home address space (ASID)

- Any address spaces involved in cross-memory mode processing
- The same processor (CPU), if the problem occurred while the system was disabled for input/output (I/O) and external interrupts (EXT), as indicated in STATUS CPU dump output

Look for the following:

- In SUMMARY TCBERROR output, look at the task completion codes in job step program TCB CMP fields; a nonzero completion code indicates an abend. You are looking at the correct abend if it has an associated RTM2WA. If a related task abended seconds before the abend being diagnosed, check the task's CDE, RTM2WA, and SDWA control blocks for the module name and other data about the abend. The output contains one RTM2WA for each abend being processed.

Example: SUMMARY TCBERROR Output

In the following output, the nonzero CMP field of the TCB indicates an error.

```
TCB: 009F3E88
   CMP...... 940C1000  PKF...... 80        LMP...... FF        DSP...... FF
   TSFLG.... 00        STAB..... 009FD200  NDSP..... 00002000
   JSCB..... 009FF40C  BITS..... 00000000  DAR...... 00
   RTWA..... 00000000  FBYT1.... 00
   Task non-dispatchability flags from TCBFLGS5:
    Secondary non-dispatchability indicator
   Task non-dispatchability flags from TCBNDSP2:
    SVC Dump is executing for another task
```

- In VERBEXIT LOGDATA output or the logrec reports and in messages in all locations, look for previous abends and symptom records for earlier problems that did not cause abends. The previous abend or the earlier problem might have led to the abend being diagnosed.
- Look for the name of the program that called the abending module.
  - The address of the calling program can be in the second half of the PSW stored in the caller's RB, which will precede the running RB, except for branch entries.
  - Determine the linkage conventions of individual save areas. The calling program's address may be in a save area.
  - In register 14 of the top RB.
  - If a command or macro was being processed, obtain the name of the module issuing the command or macro. The name is in the NAME field of the CDE for a request block (RB) for the abending module's TCB.

  Check for problems in the calling program. The calling program might have caused the abend being diagnosed.

Investigate the following:

- Many abends relating to the same area of the system
- Many TCBs with the same abend code

References

- See *z/OS MVS IPCS Commands* for information about IPCS subcommands.
- See *z/OS MVS Data Areas, Vol 1 (ABEP-DALT)* for the format and contents of the CDE.
- See *z/OS MVS Data Areas, Vol 3 (IVT-RCWK)* for the format and contents of the RB.

- See *z/OS MVS Data Areas, Vol 4 (RD-SRRA)* for the format and contents of the RTM2WA and SDWA.
- See *z/OS MVS Data Areas, Vol 5 (SSAG-XTLST)* for the format and contents of the TCB.

Step 14: Gather Additional Data, Contact IBM

Report the problem to IBM, if assistance is needed or if the problem is new.

Provide the following problem data:

- Any changes made to the system recently, preceding when the problem began occurring (for example, PTFs or new products installed or new hardware).
- Problem type: abend
- Search argument
- Dump, formatted by IPCS, online or printed
- Failing input request: macro, command, or statement
- SDWAVRA keys, lengths, and contents
- Offset of the failing instruction into the module or CSECT
- Accompanying messages: identifiers and texts
- Logrec report, if used
- All printed output and output data sets related to the problem
- Data on any related problems
- Module name and level
- Name and level of the operating system(s) with a list of program temporary fixes (PTF) applied at the time of the problem and all installation modifications, exits, and products with other than Class A service
- Other problem data developed while using the diagnosis book for the component, subsystem, or program

Reference: See “Reporting a Problem to IBM” on page 1-13 for more information.

Chapter 3. Diagnosing a System Hang

Symptoms

The system enters a wait or the entire system hangs. The terms hang and wait are used synonymously in this procedure. Some symptoms of a wait/hang are:

- No response on user's or system operator's console.
- The operator cannot communicate with the system through the console with master authority or the alternate console.
- The system does not issue messages to the console with master authority or the alternate console. The system does not receive messages from these consoles.
- The operator witnesses a series of WAIT indicators followed by a burst of activity.
- A message indicating a wait appears on the system console.

The types of hangs/waits are:

- **Disabled wait with a wait state code**: The system issues a wait state code and stops. The operator can see the wait state code on the system console. This wait is called a **coded wait state** or a **disabled wait**. There are two types of disabled wait state codes, restartable and non-restartable. For a non-restartable wait state code, the operator must reIPL the system. For a restartable wait state code, the operator may restart the system.
- **Enabled wait**: The system stops processing without issuing a wait state code when the dispatcher did not find any work to be dispatched. The operator sees a WAIT indicator on the system console, followed by a burst of activity caused by system resources manager (SRM) processing, followed by the WAIT indicator, followed by a burst of activity, and so on. An indication of an enables wait is a PSW of X'07xxxxxx xxxxxxxx'.

  A special type of enabled wait is called a **no work wait** or a **dummy wait**. An indication of a dummy wait or no work wait is a PSW of X'070E0000 00000000' and GPRs containing all zeroes. Diagnosis is required for this type of wait only when the system does not resume processing.

What You'll Need

- IPCS installed
- Stand-alone dump
- Logrec error record
- Job log

About Using the Procedure

The following waits are not covered in this procedure:

- The following coded wait states should be diagnosed using the procedure in “Chapter 2. Diagnosing an Abend” on page 2-1:
  – Wait state X'040'
  – Wait state X'064' with X'09' in the second byte of the right half of the program status word
- Wait state X'01B' occurs because of a SLIP trap that was set with an action of WAIT. You may need a stand-alone dump to diagnose the problem.

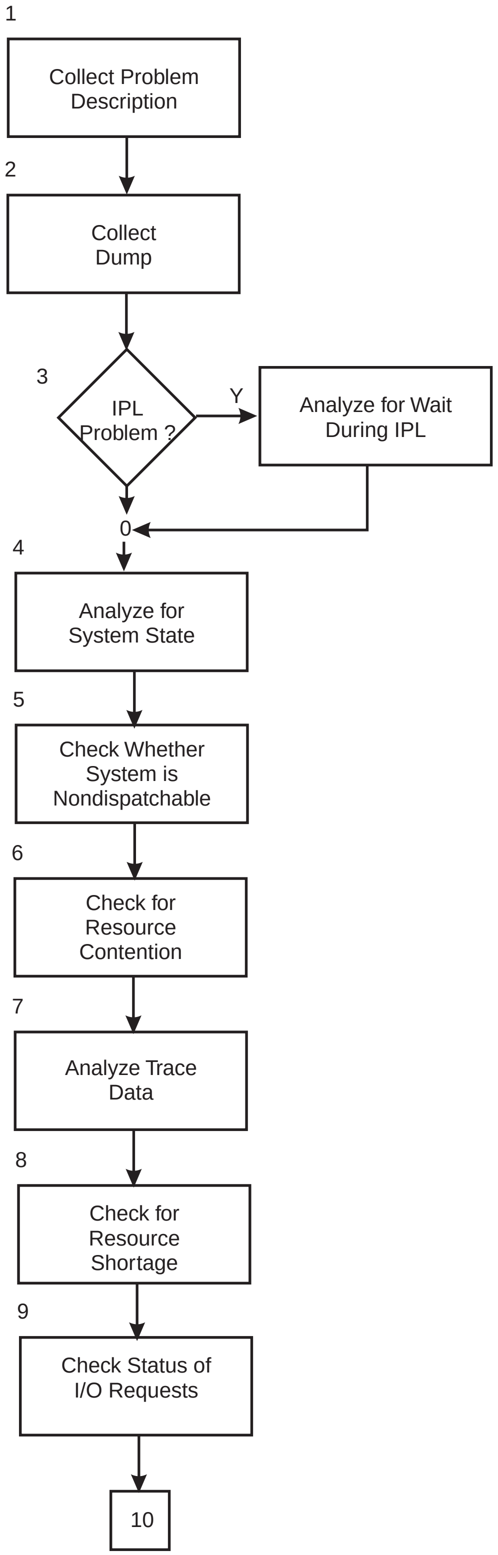
1
Collect Problem Description
2
Collect Dump
3
IPL Problem ?
Y
Analyze for Wait During IPL
0
4
Analyze for System State
5
Check Whether System is Nondispatchable
6
Check for Resource Contention
7
Analyze Trace Data
8
Check for Resource Shortage
9
Check Status of I/O Requests
10

Figure 3-1. Diagnosing a Hang or Wait (Part 1)

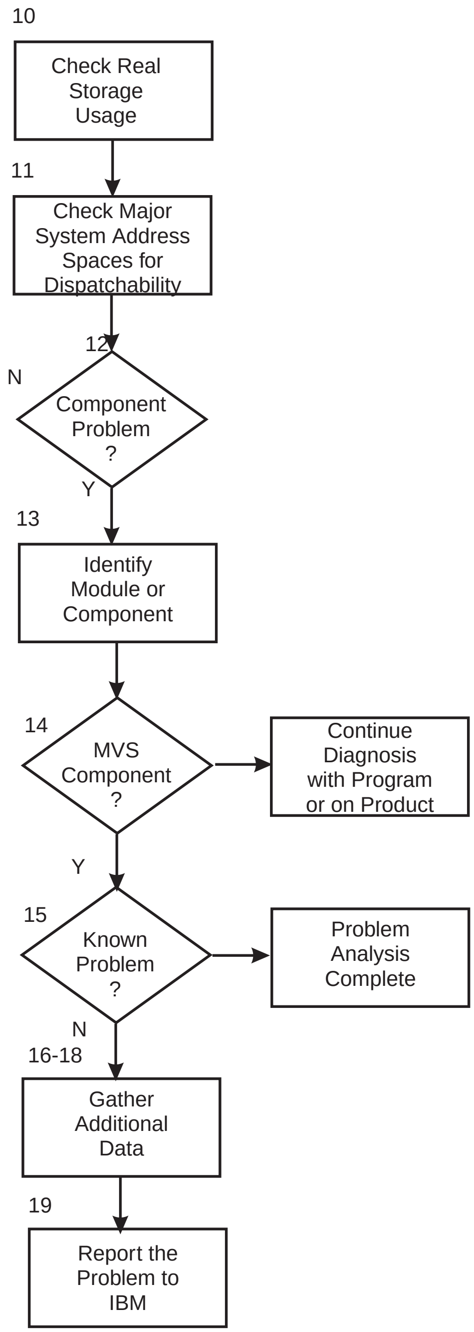
10
Check Real Storage Usage
11
Check Major System Address Spaces for Dispatchability
12
N
Component Problem ?
Y
13
Identify Module or Component
14
MVS Component ?
Continue Diagnosis with Program or on Product
Y
15
Known Problem ?
Problem Analysis Complete
N
16-18
Gather Additional Data
19
Report the Problem to IBM

Figure 3-2. Diagnosing a Hang or Wait (Part 2)

Step 1: Collect Problem Description

Ask the operator for a description of the problem.

- **Ask the operator for a description of the problem to identify the type of hang or wait.**

  Ask the operator for a description of the problem and ascertain whether you have a hang, a disabled wait state, or an enabled wait state that needs diagnosis. Use the symptom descriptions in the symptom description on page 3-1 to identify the problem.

If the operator sees in the system activity display (SAD) on the system console high system activity, at nearly 100%, and no jobs being processed, the problem is a loop. Use the procedure in “Chapter 4. Diagnosing a Loop” on page 4-1.

- **Ask if the operator could communicate with the system.**

  Ask the operator whether any of the following symptoms occurred:

  – No response on user’s or system operator’s console.

  – The operator cannot communicate with the system through the console with master authority or the alternate console.

  – The system does not issue messages to the console or receive replies from the operator.

  If the operator cannot communicate with the system, the system may be experiencing either a console I/O problem or a hung communications task (COMMTASK) component. Obtain a stand-alone dump, then do steps “Step 12: Component Problem?” on page 3-15 through “Step 19: Report the Problem to IBM” on page 3-22.

Step 2: Collect Dump

Obtain a dump for the problem.

Store status, if the system does not do this automatically. Then request a stand-alone dump.

If the operator did not obtain a dump, re-create the problem if necessary, requesting the a stand-alone dump.

Reference

See *z/OS MVS Diagnosis: Tools and Service Aids* for requesting a stand-alone dump.

Step 3: IPL Problem?

Analyze the dump for a wait or hang during IPL.

If a problem occurs during IPL, do the following:

- Look for a wait state code in one of the following places in the stand-alone dump:

  – Current program status word (PSW)

  – Machine check new PSW in the FLCMNPSW field of the prefixed save area (PSA), if a program check occurred before the recovery termination manager (RTM) was initialized

  – System wait state PSW in the NVTWTPSW field in the NIP vector table (NVT), if the system detected a problem during nucleus initialization program (NIP) processing. The CVTNVT0 field in the communication vector table (CVT) points to the NVT.

    Use an IPCS LIST subcommand to look in the NVT.

- If you don’t find a wait state code, do the following:

  1. To determine which resource initialization module (RIM) was last running, do the following:

     – Look at the CVT using the CBFORMAT CVT subcommand. In field NVT0 at offset X'4C4', find the address of the NVT.

Example: Find the address of the NVT

In the following example, the address of the NVT control block is X'00F7A360':

```
CVT: 00FD0858
   -0028  PRODN.... SP4.2.3   PRODI.... JBB4423   VERID....
   -0006  MDL...... 3090      RELNO.... 038
   +0000  TCBP..... 00000218  0EF00.... 00FED128  LINK..... 00F05FC8
   +000C  AUSCB.... 00FD0E40  BUF...... 00000000  XAPG..... 00FF05C8
   +0018  0VL00.... 00FFE8B6  PCNVT.... 00FF1EF6  PRLTV.... 00FF1D90
   +0024  LLCB..... 012D5408  LLTRM.... 8114D5E0  XTLER.... 00FF7118
                            .
                            .
                            .
   +04BC  SPIP..... 00000000  DFA...... 00FD029C  NVT0..... 00F7A360
   +04C8  CSOMF.... 00000000  CSOAL.... 00000000  ICHPT.... 01D28F00
```

– Format the NVT using the LIST *nvt_addr* subcommand with the address of the NVT found in the step above. Look in the NVT at offset X'4' for the name of the resource initialization module (RIM) active at the time of the problem. Use this RIM name as a search argument to search problem reporting databases for a fix for the problem.

2. Format the dump with the STATUS REGS subcommand to display the PSW and registers. The second half of the PSW can be useful as a pointer to the area where the problem occurred.

Example: STATUS REGS output

In the following output, the PSW contains a pointer to X'0124EE9C'.

```
CPU(X'01') STATUS:

 PSW=070C9000  8124EE9C  (RUNNING IN SECONDARY, KEY 0,  AMODE 31, DAT ON)
     DISABLED FOR PER
   NOCPU ASID(X'0001') 0124EE9C. IEANUC01.IGVSLIS1+0ADC IN READ ONLY NUCL
  ASCB1 at F56180, JOB(*MASTER*), for the home ASID
  ASXB1 at 6FDED8 and a local SRB for the home ASID
  HOME ASID: 0001 PRIMARY ASID: 0001 SECONDARY ASID: 0002

 CLTE: 01E6D188
  +0000  BLSD..... 00000000  XDS...... 00000000  XRES..... 00000000
  +000C  XQ....... 00FD4900  ESET..... 00FD4908  ULUT..... 00FD4910
  HOLDING LOCK(S): CML_for_ASID_X'0002'
  CURRENT FRR STACK IS: NORMAL

  GPR VALUES
      0-3  01D4CD58  01D4CD50  00000002  00001000
      4-7  01D4C240  7F7B3760  7F724CD0  00001000
      8-11 0124E3C0  0124F3BF  01D4C3A8  01D4C418
     12-15 01D4C418  01D4C290  8124EEEE  0000011C
```

3. Determine what program the PSW is pointing to, using the WHERE subcommand with the address in the second half of the PSW. Look for problems in the program that the PSW is pointing to.

Example: What is the PSW pointing to?

The following example shows the output from the subcommand:

```
WHERE 0124EE9C
```

The address used is from the PSW. The PSW pointed to load module IEANUC01 which is X'ADC' bytes into the CSECT IGVSLIS1.

```
NOCPU ASID(X'0001') 0124EE9C. IEANUC01.IGVSLIS1+0ADC IN READ ONLY NUCLEUS
```

References

- See *z/OS MVS Data Areas, Vol 1 (ABEP-DALT)* for the CVT.
- See *z/OS MVS Data Areas, Vol 3 (IVT-RCWK)* for the format and contents of the IVT.
- See *z/OS MVS Data Areas, Vol 3 (IVT-RCWK)* for the NVT and PSA.
- See *z/OS MVS IPCS Commands* for the LIST subcommand.

Step 4: What Kind of System State

Analyze the dump for the type of system state.

If you are unsure about what state the system is in, use this procedure to identify it. In a multiprocessor (MP) system, do this step for the dump from each processor.

Format the dump with the STATUS CPU subcommand. Under the heading CPU(X'nn') STATUS, look for one of the following:

`DISABLED WAIT STATE CODE hhh`
`REASON CODE hhhhhhhh`

The system stopped for a disabled wait. Look at the explanation of the wait state code and any accompanying reason code. Take the recommended actions.

If the wait state code does not appear in *z/OS MVS System Codes*, look at the program status word (PSW) in the STATUS CPU output. Determine if the PSW is valid; a valid PSW for a coded wait state has the following characteristics:

– Bit 14, the wait state bit, is on. That means that first half of the PSW is one of the following:
  - X'nnn2nnnn'
  - X'nnn3nnnn'
  - X'nnn6nnnn'
  - X'nnn7nnnn'
  - X'nnnAnnnn'
  - X'nnnEnnnn'
  - X'nnnFnnnn'

– The second half of the PSW has one of the following formats:
  - X'rrrrrwww'
  - X'8rrrrwww'

  www is the wait state code and *rrrrr* or *8rrrr* is a supplement code for the wait state code. The wait state determines the size and position of the supplement code. Usually the supplement code is a reason code in the PSW. If the supplement code is loaded with the LOADWAIT macro, this field begins with an 8; ignore it. The 8 is not a part of the reason code. See the reason code for the appropriate wait state code in *z/OS MVS System Codes*.

If the PSW is not valid for a coded wait state, then the system loaded a PSW that had been overlaid in the prefix save area (PSA) or in a data area, such as a request block (RB) or a save area for a request block.

Example: Disabled Wait with a Wait State Code

The following example shows a disabled wait with a valid PSW showing a wait state code of X'064'. Note that the wait bit in the PSW (bit 14) is on.

```
CPU(X'00') STATUS:

PSW=000A0000  80009064  DISABLED WAIT STATE CODE 064SUPPLMNT INFO 8000
IEA41001I NIP RIM IEAVNPC4 has failed
  ASCB1 at FD1600, JOB(*MASTER*), for the home ASID
  ASXB1 at FD1878 and a global SRB for the home ASID
  HOME ASID: 0001 PRIMARY ASID: 0001 SECONDARY ASID: 0001
```

NO WORK WAIT

The system is in an enabled wait. The system hung because it had no work waiting to be dispatched, either because the work was waiting for a resource or because no work remained on the system queues.

In a no work wait, the PSW contains X'070E0000 00000000'.

If you see NO WORK WAIT, there is probably more than one key process backlogged. Continue diagnosis with the following steps in this procedure to identify the problems and to determine which problem caused all the others.

DISABLED FOR cccccccc

System processing was disabled for one or more types of interrupts for the module running at the time of the dump. The system can be disabled for program event recording (PER), input/output (I/O), external interrupts (EXT), and machine checks (MCH).

- The system is in a disabled state if it is disabled for I/O or EXT or both. The first bye of the PSW should be something other than X'07' or X'47'.
- The system is in an enabled state if it is not disabled for I/O or EXT. The first byte of the PSW should be X'07' or X'47'.

Example: Enabled State Identified in STATUS CPU Output

In the following example, you can identify the state as an enabled one, because the first bye of the PSW is X'07':

```
 CPU STATUS:

  PSW=070C2000  83D35B2A  (RUNNING IN PRIMARY, KEY 0,  AMODE 31, DAT ON)
      DISABLED FOR PER
    ASID(X'0001') 03D35B2A. AREA(PRIVATEX)+035B2A IN EXTENDED PRIVATE
   ASCB1 at FD4680, JOB(*MASTER*), for the home ASID
   ASXB1 at FD4930 for the home ASID. No block is dispatched
   HOME ASID: 0001 PRIMARY ASID: 0001 SECONDARY ASID: 0001
```

In either state, the system can be disabled for MCH or PER or both.

When you see DISABLED for cccccccc, (except for PER), there is probably more than one key process backlogged. Continue diagnosis with the following steps in this procedure to identify the problems and to determine which problem caused all the others.

References

- See *z/OS MVS IPCS Commands* for the STATUS subcommand.
- See *z/OS MVS System Codes* for an explanation of the wait state code.
- See *Principles of Operation* for the PSW.

Step 5: Is the System Nondispatchable?

Analyze the dump to see if the system is nondispatchable.

If the operator took a stand-alone dump for a no work or dummy wait, check to see if the system was left non-dispatchable using the STATUS WORKSHEET subcommand through the CBFORMAT RTCT command.

- Format the dump with the STATUS WORKSHEET command.

Example: Viewing Dispatchability in STATUS WORKSHEET

In the following example, the dump shows that SVC dump left the system nondispatchable:

```
                MVS Diagnostic Worksheet
                    .
                    .
                    .

  SYSTEM RELATED DATA

  CVT  SNAME (154) SYSTEM1    VERID (-18)

       CUCB   (64) 00FD5FC8   PVTP  (164) 00FE17A0    GDA  (230) 01F17170

       RTMCT (23C) 00FC6780   ASMVT (2C0) 00FD9490    RCEP (490) 012E7810

  CSD Available CPU mask: 1800 Alive CPU mask: 1800 No. of active CPUs: 0002

       System set non-dispatchable by SVC dump
```

- If they do not, you probably have an SDUMP or RTM problem. Contact the IBM Support Center.
- If the system is non-dispatchable, you may have an SVC or RTM problem. Check the logrec data set and SYSLOG for errors from around the time of the dump. Then contact the IBM Support Center.

References

- See *z/OS MVS IPCS Commands* for the CBFORMAT and STATUS subcommands.
- See *z/OS MVS Data Areas, Vol 4 (RD-SRRA)* for a map of RTCT.

Step 6: Check for Resource Contention

Find the resource causing an enabled wait or hang.

If you have an enabled wait (a wait without a wait state code), format the dump with the ANALYZE EXCEPTION subcommand. ANALYZE performs contention analysis of suspend locks, queues, I/O devices, page waits, and device allocation waits. If work was waiting for a resource, the ANALYZE output identifies the job, address space (ASID), and task (TCB) that held the resource. If no work was waiting, continue diagnosis with the next step.

Example: ANALYZE EXCEPTION Sample Output

In the following example, there are 61 units of work waiting for resource #11:

```
                 CONTENTION EXCEPTION REPORT

 JOBNAME=*MASTER*  ASID=0001  TCB=006E8E88

 JOBNAME=*MASTER* HOLDS THE FOLLOWING RESOURCE(S):

    RESOURCE #0011:  There are 0061 units of work waiting for this resource.
       NAME=MAJOR=SYSIEA01 MINOR=DMPDSENQ SCOPE=SYSTEM

  STATUS FOR THIS UNIT OF WORK:
    IRA10102I This address space is on the SRM IN queue.
    Task non-dispatchability flags from TCBFLGS4:
     Top RB is in a wait
```

To find out more about the resource that you isolated in the ANALYZE EXCEPTION output, use the ANALYZE CONTENTION subcommand.

If you find a resource contention problem and have found the source of the problem, identify the program being processed when the hang or wait occurred by issuing a SUMMARY FORMAT subcommand to find the jobname. Find also the unit of work holding the resource and examine its status. Then continue diagnosis with “Step 13: Identify Module or Component” on page 3-16.

If you have a resource contention problem, but you cannot pinpoint the cause, then continue diagnosis with “Step 12: Component Problem?” on page 3-15.

References

- See *z/OS MVS IPCS Commands* for the ANALYZE and STATUS IPCS subcommands.
- See *z/OS MVS Diagnosis: Reference* for ENQ resource names.
- See *z/OS MVS JCL Reference* for the EXEC statement.

Step 7: Analyze Trace Data

Analyze the system trace table.

Format the dump with a SYSTRACE subcommand to analyze the system trace table. Start with the last entry, and back up through the trace table. Look for one or more of the following:

- **A loop, indicated by:**
  – Repeated requests by a program for one or more system services, that is, repeated CLKC or DSP entries.

Example: Loop in SYSTRACE Output

In the following output, the repeated CLKC entries indicate a loop:

```
01 003E 00000000   CLKC      070C0000 8100765C  00001004 00000000
01 003E 00000000   CLKC      070C2000 81005638  00001004 00000000
01 003E 00000000   CLKC      070C0000 810056E6  00001004 00000000
01 003E 00000000   CLKC      070C0000 80FF0768  00001004 00000000
01 003E 00000000   CLKC      070C0000 80FE4E34  00001004 00000000
01 003E 00000000   CLKC      070C1000 81004BB8  00001004 00000000
```

- The same task (TCB-ADDR address) in most trace entries. This could be either normal operation or a loop in a task.
- Presence of trace entries for the address space identified with the problem.

  Format the dump with a SYSTRACE ’ASID(*asid*)’ where *asid* is the number for the address space associated with the job which is hung or causing the contention on the system. You should have identified the job and address space in “Step 6: Check for Resource Contention” on page 3-9.

  If there are trace entries for this address space, this indicates a possible loop. Note the PSW addresses from several of the trace entries and use them in a WHERE subcommand to determine what module is executing.

An enabled loop has multiple types of entries before the pattern repeats. Continue diagnosis with “Chapter 4. Diagnosing a Loop” on page 4-1.

- **A stop entered by the operator.**

  A stop is indicated by a large gap in the time stamps. If the stop is not followed by a restart, indicated by *RST in the IDENT column, discontinue diagnosis and tell operations.

References

- See *z/OS MVS IPCS Commands* for the SYSTRACE subcommand.
- See *z/OS MVS Diagnosis: Tools and Service Aids* for the format of system trace entries and analyzing a system trace.
- See *z/OS MVS Programming: Assembler Services Reference IAR-XCT* for the WAIT and POST macros.

Step 8: Check for Resource Shortage

Look for a program using an extremely large amount of system resources.

Look at one of the following for a program using a large amount of system resources:

- RMF reports
- SMF reports
- Output from another system monitoring program
- The job log

If a program used a large amount of system resources, the cause is probably a loop in the program. Continue diagnosis with “Chapter 4. Diagnosing a Loop” on page 4-1.

References

- See *z/OS MVS System Management Facilities (SMF)* for the SMF reports.
- See *RMF User’s Guide* for information about using RMF.

Step 9: Check the Status of I/O Requests

Analyze outstanding I/O requests for problems.

Format the stand-alone dump with the IOSCHECK ACTVUCBS subcommand to display the IOS control block and active UCBs. Use the output to determine the status of I/O activity when the system entered the wait state.

Look for the following in the ACTVUCBS Processing section of the output:

- HOTIO field - indicator for a hot I/O condition. A non-zero value indicates that there is a hot I/O condition.
- IOQF and IOQL fields - show the address of the first (IOQF) and last (IOQL) I/O request for the device associated with the UCB.

Example: IOSCHECK ACTVUCBS Subcommand Output

In the following output, the HOTIO field indicates that a solicited interrupt has completed with other than DCC-3 since the last time HOT-I/O detection was called. Note also that the IOQF and IOQL fields are identical, indicating that the first and last request for this device is the same.

```
                      * * * ACTVUCBS   Processing * * *
UCB AT 00F8B798:  DEVICE  001; SUBCHANNEL 0001

 UCBPRFIX: 00F8B768
    -0030  RSTEM.... 00        RSV...... 08        MIHTI.... 40
    -002D  HOTIO.... 40        IOQF..... 00F7BC00  IOQL..... 00F7BC0

    -0024  SIDA..... 0001      SCHNO.... 0001      PMCW1.... 2888
    -001E  MBI...... 0000      LPM...... 80        RSV...... 00
    -001A  LPUM..... 80        PIM...... 80        CHPID.... 21000000
    -0014            00000000  LEVEL.... 01        IOSF1.... 00
    -000E  MIHCT.... 0000      LVMSK.... 00000001  LOCK..... 00000000
    -0004  IOQ...... 00F7BC00
```

Step 10: Check Real Storage Usage

Analyze real storage usage for problems.

Do the following:

- Format the stand-alone dump with the RSMDATA subcommand to obtain information about real storage usage and any unusual conditions occurring at the time of the problem. If the percent usage field is 100%, the problem is that there are no real frames left. If the percent of available total fixed frames is high (above 50%), there may be a program that is failing to complete and release the resources.

Example: RSMDATA Output

In the following output, the percent usage of real storage is at 99% and the percent of available total fixed frames is at 25%.

```
                     R S M   S U M M A R Y   R E P O R T

                          Tot real Below Prf real Dbl real      Expanded
                          -------- ----- -------- -------- -------------
In configuration . . . .    33,792 4,096   33,742        -        49,152
Available for allocation    32,672 4,089   33,742      120        49,152
Allocated  . . . . . . .    32,398 3,964   33,483      113        48,594
Percent usage  . . . . .        99    96       99       94            98
Common fixed frames  . .     3,087   317    3,087        -             -
  Percent of available .         9     7        9        -             -
Total fixed frames . . .     8,338   907        -        -             -
  Percent of available .        25    22        -        -
  -
```

- Check the ASM control blocks to determine the statistics applicable to I/O requests by formatting the stand-alone dump with ASMCHECK. The number of I/O requests received and completed should be identical or very close.

Example: ASMCHECK Output

In the following output, note that the 509577 I/O requests received have all been completed.

```
ASMVT AT 00FD8030
509577 I/O REQUESTS RECEIVED, 509577 I/O REQUESTS COMPLETED BY ASM
240487 NON-SWAP WRITE I/O REQUESTS RECEIVED, 240487 NON-SWAP WRITE I/O
       REQUESTS COMPLETE
PART AT 01CB5310
PAGE DATA SET 0 IS ON UNIT 15B
PAGE DATA SET 1 IS ON UNIT 15B
PAGE DATA SET 3 IS ON UNIT 14A
PAGE DATA SET 4 IS ON UNIT 150
PAGE DATA SET 5 IS ON UNIT 15B
```

Step 11: Check Major System Address Spaces for Dispatchability

Analyze major system address spaces to determine if they are dispatchable.

By performing address space analysis on the major system address spaces, you can determine if there is any work waiting and if the address space is dispatchable. The major address spaces you should analyze are:

- Master scheduler, ASID 1
- Console
- JES2/JES3
- IMS/CICS/VTAM

Format the stand-alone dump with the SUMMARY FORMAT subcommand and examine the ASCBs and TCBs to analyze an address space for dispatchability. Refer to the SUMMARY subcommand in *z/OS MVS IPCS Commands* for examples of the output provided by SUMMARY FORMAT.

WEBs representing TCBs and SRBs which are dispatchable can be found on the system work unit queue and the processor related work unit queues.

SRBs can also be found on the global service manager queue and the local service manager queue.

- Are there any ready TCBs? Ready TCBs are indicated by fields:
  – ASCBTCBS - number of ready TCBs plus the number of TCBs requiring the local lock.
  – ASCBTCBL - number of TCBs requiring the local lock.
- If there is ready work, check to see if the ASCB is dispatchable. Look at field ASCBDSP1, the non-dispatchability flag. See *z/OS MVS Data Areas, Vol 1 (ABEP-DALT)* for what the values of ASCBDSP1 indicate.

If ASCB analysis indicated that there was no ready work available to be dispatched, look at the TCBs to determine if they are in a normal wait.

In the TCB, look for:

- If there is no ready work, are the TCBs in a normal wait? A non-zero value in fields TCBFLGS4, TCBFLGS5, or TCBNDSP indicates that the TCB is non-dispatchable. At the end of the TCB format, the dump will also contain a statement if the TCBFLGS4, TCBFLGS5, or TCBNDSP bits are on.
- Find the address of the current request block (RB), pointed to by field TCBRBP.

Example: TCB in SUMMARY FORMAT Output

In the following output, the top RB is in a wait. This is indicated by field FLGS and by the statement at the end of the TCB.

Note also that address 006FF048 in field RBP contains the address of the current RB.

```
TCB: 00FD3608
   +0000  RBP...... 006FF048  PIE...... 00000000  DEB...... 00000000
   +000C  TIO...... 00000000  CMP...... 00000000  TRN...... 00000000
   +0018  MSS...... 7F7463A0  PKF...... 00        FLGS..... 00008004  00
   +0022  LMP...... FF        DSP...... FF        LLS...... 006FFD38
   +0028  JLB...... 00000000  JPQ...... 006FF200
 GENERAL PURPOSE REGISTER VALUES
     0-3  00000001  000027C4  00009FBC  00000004
     4-7  006FFF48  006FEFB8  00F6E900  0000005C
     8-11 80001E52  00C0DCE8  006F5FF0  00FCF778
    12-15 00FCF170  006FF348  80FCF1C0  806FF048
   +0070  FSA...... 00000000  TCB...... 006FF6F0  TME...... 00000000
   +007C  JSTCB.... 00FD3608  NTC...... 00000000  OTC...... 00000000
   +0088  LTC...... 006FF6F0  IQE...... 00000000  ECB...... 00000000
   +0094  TSFLG.... 00        STPCT.... 00        TSLP..... 00
   +0097  TSDP..... 00        RD....... 7F748F04  AE....... 7F746280
   +00A0  STAB..... 00F0B860  TCT...... 00000000  USER..... 00000000
   +00AC  NDSP..... 00000000  MDIDS.... 00000000  JSCB..... 00C0BE84
        .
        .
        .
   +014C  BDT...... 00000000  NDAXP.... 00000000  SENV..... 00000000
   Task non-dispatchability flags from TCBFLGS4:
    Top RB is in a wait
```

In the RB, look for:

- Field RBOPSW, containing the user's old PSW.

Example: OPSW Field in SUMMARY FORMAT Output

In this example, the address of the module holding the resource is X'1FC08E4', and is found in the right half of the PSW in the OPSW field of the request block following the TCB that held the resource.

```
ACTIVE RBS
 PRB: 009F81C0
    -0020  XSB...... 009F8228  FLAGS2... 00000080  RTPSW1... 00000000
    -0014            00000000  RTPSW2... 00000000  0015D001
    -0008  FLAGS1... 02000005  WLIC..... 00020001  SZSTAB... 00110082
    +000C  FLCDE.... 009F8BA8  OPSW..... 070C6000  81FC08E4
    +0018  SQE...... 00000000  LINK..... 009F82C0
    +0020  GPR0-3... 7FFEA5E0  07D04D78  07D31000  07D40FFF
    +0030  GPR4-7... 00000000  81099110  00000028  07D04AD0
    +0040  GPR8-11.. 07D038FD  07D028FE  07D018FF  07D045A0
    +0050  GPR12-15. 87D00900  00000000  00F976F0  07D04D18
```

Use the address from the PSW found in the SUMMARY FORMAT output for the RB in a WHERE *address* subcommand to identify the waiting program.

If the program resides on in link pack area (LPA) or private storage use the AMBLIST service aid to identify the CSECT. See *z/OS MVS Diagnosis: Tools and Service Aids* for information about using AMBLIST.

If it is a system component causing the hang or wait, See "Step 12: Component Problem?" to analyze component status.

Step 12: Component Problem?

Analyze system components or subsystems for problems.

Hangs or Waits can be caused by system or subsystem problems, For example, a storage problem can hang a system. Format the stand-alone dump with the following IPCS subcommands and look in the output for indications of problems:

```
ASMCHECK
COMCHECK
COUPLE SUMMARY ALL
COUPLE DETAIL ALL
COUPLE EXCEPTION ALL
DIVDATA SUMMARY ALL
DLFDATA
IOSCHECK
RSMDATA SUMMARY
RSMDATA EXCEPTION
VERBEXIT ALCMWAIT
VERBEXIT AOMDATA
VERBEXIT ASMDATA
VERBEXIT AVMDATA
VERBEXIT GRSTRACE
VERBEXIT LOGDATA
VERBEXIT MMSDATA
VERBEXIT JES2 or VERBEXIT JES3
VERBEXIT SMSDATA
VERBEXIT SRMDATA
VERBEXIT TSODATA
VERBEXIT VSMDATA
VLFDATA
```

Look in VERBEXIT LOGDATA output for records about I/O problems, and continue diagnosis with any I/O problem.

System Hang Analysis

Look in COMCHECK output for errors on console devices. This can be helpful to determine why the operator could not communicate with the system.

For teleprocessing (TP) network problems, add the following IPCS subcommands. Problems in the TP network often appear as a waiting network or as waiting terminals, even as waiting systems.

```
VERBEXIT VTAMMAP
VERBEXIT TCAMMAP
```

If you find a system component problem, continue diagnosis with “Step 13: Identify Module or Component”. Otherwise, if you do not find a system component problem, continue diagnosis with “Step 15: Known Problem?” on page 3-18.

References

- See *z/OS MVS IPCS Commands* for the subcommands.
- For problems indicated in the IPCS output for a subsystem or product, see the book for the subsystem or product:
  - *z/OS JES2 Diagnosis*
  - *z/OS JES3 Diagnosis*
  - *z/OS DFSMSdfp Diagnosis Reference*
  - *z/OS Communications Server: SNA Diagnosis V1 Techniques and Procedures*
  - *z/OS Communications Server: SNA Diagnosis V2 FFST Dumps and the VIT*
  - *z/OS MVS Diagnosis: Reference* for the format of logrec records or diagnostic information concerning an MVS component.

Step 13: Identify Module or Component

Identify the module, component, subsystem, or product that caused the hang or wait.

- **For a coded wait state:** Use the explanation for the code to identify the module, component, or subsystem.

Example: Explanation of the Code

In the following example, the explanation of the code in *z/OS MVS System Codes* identifies the component as Input/output supervisor (IOS).

079

Explanation: The system could not recognize the I/O configuration.

Source: Input/output supervisor (IOS)

System Action: The system places message IEA098W in the initial program load (IPL) write to operator (WTO) buffer to identify the incompatible I/O configuration. The system enters wait state X’079’

Operator Response: Request a stand-alone dump. Then reIPL the system, using a different nucleus member, if possible.

System Programmer Response: Obtain the I/O configuration identifier by locating message IEA098W in the IPL WTO buffer. ReIPL the system with an I/O configuration that is compatible with the system code.

- **For a system disabled for interrupts:** Obtain the address from the right half of the PSW in the STATUS CPU output or in the last system trace entry obtained in “Step 7: Analyze Trace Data” on page 3-10.
- **For a no work wait:** Use the ANALYZE output from “Step 8: Check for Resource Shortage” on page 3-11 to identify the job, address space, and TCB holding the resource causing the wait. Find the resource-holding TCB in SUMMARY FORMAT output, then:
  - If a PRB follows the TCB, the NAME field of the contents directory entry (CDE) following the PRB may identify the module holding the resource.
  - If not, obtain the address in the right half of the PSW in the OPSW field of the request block (RB) following the TCB.

Example: OPSW Field in SUMMARY FORMAT Output

In this example, the address of the module holding the resource is X'1FC08E4', and is found in the right half of the PSW in the OPSW field of the request block following the TCB that held the resource.

```
ACTIVE RBS
 PRB: 009F81C0
    -0020  XSB...... 009F8228  FLAGS2... 00000080  RTPSW1... 00000000
    -0014            00000000  RTPSW2... 00000000  0015D001
    -0008  FLAGS1... 02000005  WLIC..... 00020001  SZSTAB... 00110082
    +000C  FLCDE.... 009F8BA8  OPSW..... 070C6000  81FC08E4
    +0018  SQE...... 00000000  LINK..... 009F82C0
    +0020  GPR0-3... 7FFEA5E0  07D04D78  07D31000  07D40FFF
    +0030  GPR4-7... 00000000  81099110  00000028  07D04AD0
    +0040  GPR8-11.. 07D038FD  07D028FE  07D018FF  07D045A0
    +0050  GPR12-15. 87D00900  00000000  00F976F0  07D04D18
```

  - Issue the WHERE *address* subcommand using the address found above in the SUMMARY FORMAT output to identify the CSECT and module.
- **For a system component that is causing the hang or wait:** Use the IPCS output from “Step 12: Component Problem?” on page 3-15 to identify the component, subsystem, or product.
- **For any problem, if the dump is an SVC dump:** Use the STATUS WORKSHEET subcommand to obtain the dump title. Obtain the component name, component identifier, or module name from the title.

Example: STATUS WORKSHEET Output

In the following output, the dump title shows the component name, XES, the component id, SCIXL, and the product ID, 5752.

```
                              MVS Diagnostic Worksheet

 Dump Title: COMPON=IXL,COMPID=5752SCIXL,ISSUER=IXLM1REC,MODULE=IXLM2SP
             ,ABEND=S0071,REASON=00000030

 CPU Model 9021 Version A6 Serial no. 300359 Address 03
 Date: 03/30/93     Time: 10:32:38  Local

 Original dump dataset: SYS1.DUMP32
```

- **To identify the module from any PSW address**, do the following. Ignore the leftmost bit in the second word of the PSW; the leftmost bit in the leftmost digit denotes addressing mode and is not part of the address.
  - If analyzing the dump interactively, use the address in a WHERE subcommand to obtain the name of the load module.
  - If analyzing printed output, find the address:

  - In dump output from the LIST or VERBEXIT SUMDUMP subcommand. Look for the CSECT name eye-catcher. IBM module names are generally followed by an assembly date and a product identifier or PTF level, such as HBB4410 or UY01234; most eye-catchers are at the beginning of a module, but some are at the end.
  - Examine the LPAMAP, NUCMAP or job pack area queue in SUMMARY FORMAT output to see which module the PSW address resides in. If it is in LPAMAP or the job pack area, the module name given will be the load module name. Use AMBLIST to identify the CSECT. See *z/OS MVS Diagnosis: Tools and Service Aids* for information on AMBLIST.
- **From the module prefix, identify the component, subsystem, or product,** if provided by IBM. See *z/OS MVS Diagnosis: Reference* to find the IBM component, subsystem, or product for a program identifier or module prefix

References

- See *z/OS MVS System Codes* for the explanation of the wait state code.
- See *z/OS MVS Diagnosis: Reference* to find the IBM component, subsystem, or product for a program identifier or module prefix.
- See *z/OS MVS IPCS Commands* for the subcommands:
  ANALYZE
  LIST
  LPAMAP
  STATUS
  SUMMARY
  VERBEXIT NUCMAP
  VERBEXIT SUMDUMP

Step 14: z/OS Component?

Choose a course of action, depending on whether this is an z/OS component.

- For a component of z/OS, continue with the next step.
- For an IBM subsystem or product, continue diagnosis with the subsystem or product.
- For an installation-provided program, including an installation exit routine, continue diagnosis with that program, using the dump.

Reference

See *z/OS MVS Diagnosis: Tools and Service Aids* for analyzing SVC, SYSMDUMP, SYSABEND, or SYSUDUMP dumps to find a problem in an installation provided program.

Step 15: Known Problem?

Search problem reporting data bases to determine if the problem was previously reported.

Use the argument under SEARCH ARGUMENT ABSTRACT in:
- Logrec detail edit report
- STATUS FAILDATA output
- VERBEXIT LOGDATA output

Example: Search Argument Abstract in VERBEXIT LOGDATA Output

The following VERBEXIT LOGDATA output contains the search argument you would use to search problem reporting data bases for a fix for the problem:

```
SEARCH ARGUMENT ABSTRACT

  PIDS/5752SC1CV RIDS/IEANUC01#L RIDS/IGC046 AB/U1409 REGS/0E05A
  REGS/0D2F8 RIDS/IGC046#R

  SYMPTOM             DESCRIPTION
  -------             -----------
  PIDS/5752SC1CV      PROGRAM ID: 5752SC1CV
  RIDS/IEANUC01#L     LOAD MODULE NAME: IEANUC01
  RIDS/IGC046         CSECT NAME: IGC046
  AB/U1409            USER ABEND CODE: 1409
  REGS/0E05A          REGISTER/PSW DIFFERENCE FOR R0E: 05A
  REGS/0D2F8          REGISTER/PSW DIFFERENCE FOR R0D: 2F8
  RIDS/IGC046#R       RECOVERY ROUTINE CSECT NAME: IGC046
```

If an argument is not provided, use the primary symptom string in VERBEXIT SYMPTOM output, if available, or use the following symptoms:

- Program identifier: PIDS/ccccccccc
- CSECT name or module name: RIDS/cccccccc
- Wait state:
  If a disabled wait, with a wait state code: WS/D0hhh
  If an enabled wait: WS/E0000
- If ANALYZE EXCEPTION output indicates a lockout: PCSS/LOCKOUT
- Input request (call, command, macro, statement), if one is associated with the problem: PCSS/cccccccccc
- Symptoms created from information in the STATUS CPU output

Select the problem type on the search tool panel, based on STATUS CPU output:

Problem Type	STATUS CPU Output
Disabled wait	DISABLED WAIT STATE CODE
Enabled wait	NO WORK WAIT
Enabled wait	DISABLED FOR ccc (not I/O or EXT)
Hang	None of the above

If the search finds no match, you can remove some symptoms or change the symptoms and search again. Continue searching for matches by adding, removing, and changing symptoms.

If the search finds that the problem was previously reported, request the problem fix. If not, continue with the next step. Use problem data from following steps to create more symptoms; use these symptoms in later searches.

References

- See “Searching Problem Reporting Databases” on page 1-4 for more information on developing search arguments.
- See *z/OS MVS IPCS Commands* for the subcommands:
  ANALYZE
  STATUS

VERBEXIT SYMPTOM
STATUS CPU

Step 16: Gather Additional Data, Messages and Logrec

Collect and analyze messages and logrec records about the problem.

Use time stamps to select messages and software, symptom, and hardware logrec records related to the problem. Look in the following:

- The job log
- A TSO/E user's ISPF transaction log or session manager log
- The hardcopy log (SYSLOG)
- VERBEXIT MTRACE dump output, which shows the buffer for system messages
- VERBEXIT LOGDATA dump output, which formats the logrec buffer
- Logrec data set, formatted by EREP

A coded wait state, a system disabled for I/O, or an I/O problem can prevent communication with an MCS console. When the system cannot communicate with an MCS console, the system places messages on the system console and in the logrec buffer.

References

- See *z/OS MVS IPCS Commands* for the VERBEXIT LOGDATA and VERBEXIT MTRACE subcommands.
- See *EREP User's Guide* for formatting logrec records.
- See *z/OS MVS Diagnosis: Reference* for logrec reports.
- For explanations of the messages, see:
  - *z/OS MVS System Messages, Vol 1 (ABA-AOM)*
  - *z/OS MVS System Messages, Vol 2 (ARC-ASA)*
  - *z/OS MVS System Messages, Vol 3 (ASB-BPX)*
  - *z/OS MVS System Messages, Vol 4 (CBD-DMO)*
  - *z/OS MVS System Messages, Vol 5 (EDG-GFS)*
  - *z/OS MVS System Messages, Vol 6 (GOS-IEA)*
  - *z/OS MVS System Messages, Vol 7 (IEB-IEE)*
  - *z/OS MVS System Messages, Vol 8 (IEF-IGD)*
  - *z/OS MVS System Messages, Vol 9 (IGF-IWM)*
  - *z/OS MVS System Messages, Vol 10 (IXC-IZP)*
  - *z/OS MVS Dump Output Messages*
  - The message book for a subsystem or program

Step 17: Gather Additional Data, Using IPCS

Collect dump data.

Format the dump and print it or store it in a data set. Use IPCS subcommands in the order indicated by the following list. If using IPCS interactively for an SVC dump, respond yes to the IPCS message that asks if summary data can be used by dump access.

```
STATUS FAILDATA
STATUS SYSTEM
STATUS CPU REGISTERS DATA CONTENTION
STATUS WORKSHEET
ANALYZE
SUMMARY TCBERROR
SUMMARY FORMAT
VERBEXIT LOGDATA
VERBEXIT SUMDUMP
SYSTRACE
VERBEXIT MTRACE
Subcommands used in (Step 12)
VERBEXIT SYMPTOM
```

References

- See *z/OS MVS Diagnosis: Tools and Service Aids* for requesting and printing dumps.
- See *z/OS MVS IPCS Commands* for the IPCS subcommands.

Step 18: Gather Additional Data, Multiple Problems

Analyze the dump data for multiple problems.

Use the dump data collected in “Step 17: Gather Additional Data, Using IPCS” on page 3-20 to search for multiple problems. Look for problems in the related data, which includes the following:

- The same job step
- The same task (TCB) or service request (SRB)
- The same home address space (ASID)
- Any address spaces involved in cross-memory mode processing
- The same processor (CPU), if the problem occurred while the system was disabled for input/output (I/O) and external interrupts (EXT), as indicated in the STATUS CPU dump output

Related data or problems will have ERRORIDs that are close. Time stamps for related problems will be a few seconds more or less than the time stamp in the last system trace entry or the last task dispatch (DSP) or service request block dispatch (SRB).

Look for the following in the dump output:

- In SUMMARY TCBERROR output, look at the task completion codes in the TCB CMP fields; a nonzero completion code indicates an abend. If a related task abended seconds before the hang or wait being diagnosed, check the task’s CDE, RTM2WA, and SDWA control blocks for the module name and other data about the abend. The output contains one RTM2WA for each abend being processed.
- In VERBEXIT LOGDATA output or the logrec reports and in messages in all locations, look for previous abends and for symptom records that might be related. The previous abend or the earlier problem might have led to the hang or wait being diagnosed.
- Look in the system trace in SYSTRACE ’ASID(*xxxx*) ’ output for RCVY entries or problem entries which have an asterisk (*) before the identifier in the IDENT column. These indicate errors that occurred previously.

Investigate the following:

- Many abends relating to the same area of the system
- Many TCBs with the same abend code

References

- See *z/OS MVS IPCS Commands* for the subcommands:
  CBFORMAT
  LIST

```
STATUS
SUMMARY
VERBEXIT LOGDATA
VERBEXIT SUMDUMP
SYSTRACE
WHERE
```

- See *EREP User's Guide* for formatting logrec records.
- See *z/OS MVS Diagnosis: Reference* for logrec reports.
- See *z/OS MVS Data Areas, Vol 1 (ABEP-DALT)* for the format and contents of the CDE.
- See *z/OS MVS Data Areas, Vol 3 (IVT-RCWK)* for the format and contents of the RB.
- See *z/OS MVS Data Areas, Vol 4 (RD-SRRA)* for the format and contents of the RTM2WA and SDWA.
- See *z/OS MVS Data Areas, Vol 5 (SSAG-XTLST)* for the format and contents of the TCB.

Step 19: Report the Problem to IBM

Report the problem to IBM, if assistance is needed or if the problem is new.

Provide the following problem data:

- Problem type: disabled wait, enabled wait, or hang
- Search argument
- Dump, formatted by IPCS, online or printed
- SMF records, if obtained
- Accompanying messages: identifiers and texts
- Hard-copy log, beginning 15 to 30 minutes before the problem, or master trace, if not wrapped between the problem and dump
- Logrec records, beginning 15 to 30 minutes before the problem and edited using the SPOTCHK and TIMESEQ parameters
- All printed output and output data sets related to the problem
- Data on any related problems
- Module name and level
- Name and level of the operating system(s) with a list of program temporary fixes (PTF) applied at the time of the problem and all installation modifications, exits, and products with other than Class A service
- Other problem data developed while using the diagnosis book for the component, subsystem, or program

References

- See "Reporting a Problem to IBM" on page 1-13 for more information on reporting problems.
- See *EREP User's Guide* for logrec formatting.

Chapter 4. Diagnosing a Loop

The two types of loops are:

- **Disabled loop** in code disabled for interrupts.

  A disabled loop usually indicates a software problem, but can indicate a hardware problem. In a few cases, the system intentionally uses a disabled loop; these cases are identified in this procedure. The PSW for a disabled loop begins with X'04'.
- **Enabled loop** in code enabled for interrupts.

  An enabled loop can include disabled portions, especially if interrupts caused the loop. The PSW for an enabled loop begins with X'07'.

Symptoms of a Disabled Loop

- There is high system activity, near 100%, that cannot be attributed to a high priority job.
- The operator is unable to communicate with the system through the master and alternate consoles.

Symptoms of an Enabled Loop

- The operator reports a bottleneck, indicating that the system slows down periodically, thus creating a performance problem.
- A job stays in the system for a long time without changing status or ending.

What You’ll Need for a Loop

- IPCS installed
- Instruction address trace
- An SVC, ABEND, or stand-alone dump
- Logrec error record
- System trace

Loop Analysis

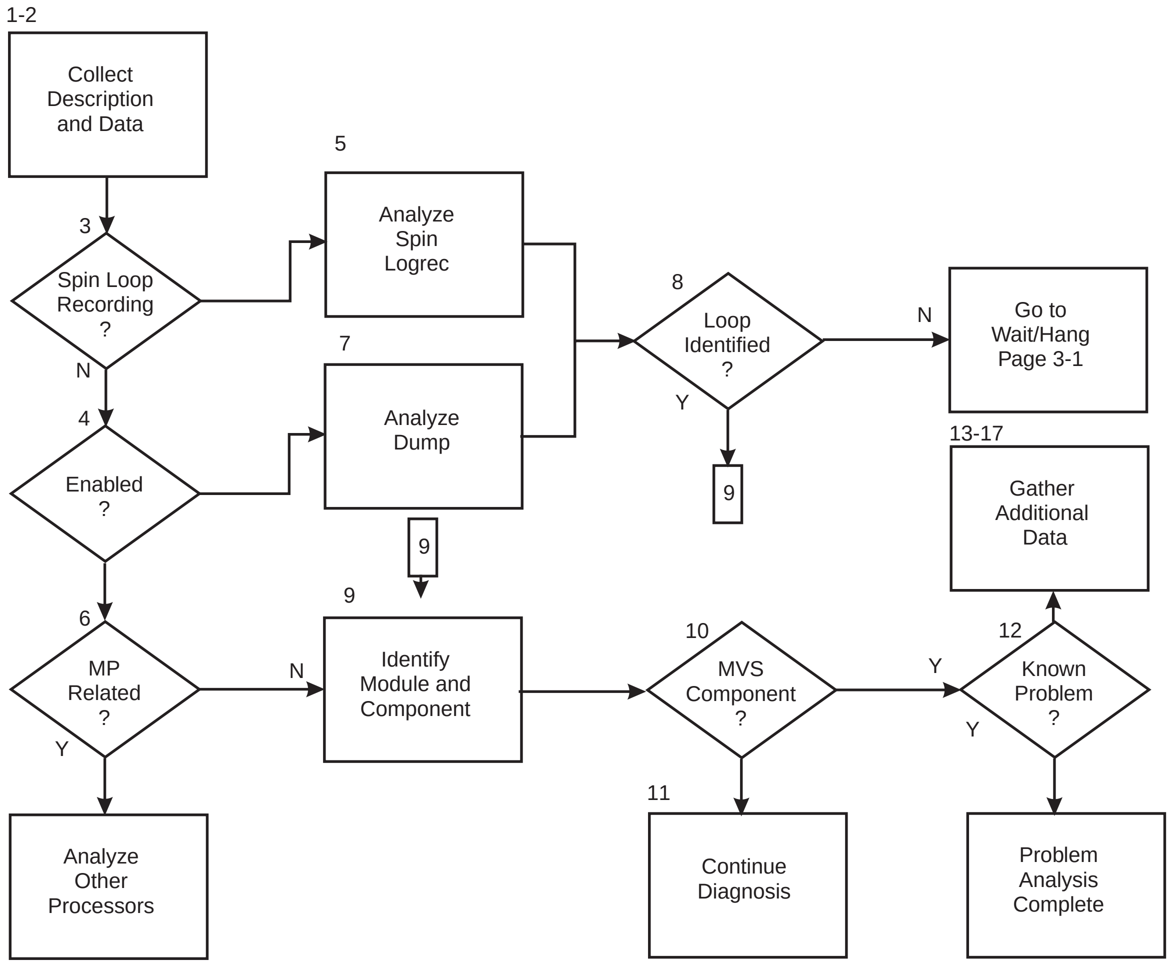

Figure 4-1. Diagnosing a Loop

Step 1: Collect Description

Ask the operator for a description of the problem.

If the operator reports high CPU activity on any of the processors, there might be a loop.

If the operator sees a WAIT indicator on the system console, followed by a burst of activity, followed by the WAIT indicator, followed by a burst of activity, and so on, the system could be in an enabled wait. Use the procedure in “Chapter 3. Diagnosing a System Hang” on page 3-1.

If the operator uses a CANCEL command to cancel a looping job, the system ends the job step with system completion code X'122' or X'222'.

Step 2: Collect Data

Collect initial problem data.

Obtain the dump requested by the operator or the system.

- For a disabled loop, the operator should request an instruction address trace, store status, and then request a stand-alone dump.
- For an enabled loop, the operator can use the CANCEL command with the DUMP option to obtain a user dump (SYSUDUMP, SYSABEND, or SYSMDUMP) or the DUMP command to obtain an SVC dump without needing to reIPL. If the SVC dump does not contain the needed data, try to specify the missing area using the SDATA operand of the DUMP command. If the SVC dump still does not contain the needed data, obtain a stand-alone dump.

References

- See *z/OS MVS Diagnosis: Tools and Service Aids* for information about stand-alone dumps, SVC dumps, and SYSMDUMP dumps.
- See *z/OS MVS JCL Reference* for the SYSMDUMP DD statement.
- See *z/OS MVS System Commands* for information about the DUMP and CANCEL commands.

Step 3: Spin Loop Recording?

Analyze for automated spin loop recovery.

Do one of the following:

- Look in a logrec error record for abend X'071'.
- Format an SVC dump or SYSMDUMP dump with a VERBEXIT LOGDATA subcommand.

Example: VERBEXIT LOGDATA Output

In the following output, the system abend code is X'071', indicating that a loop condition has occurred.

```
SEARCH ARGUMENT ABSTRACT

  PIDS/5752SC1CM RIDS/IEANUC01#L RIDS/IEAVTEXS AB/S0071 PRCS/00000010
  REGS/0E68E REGS/0C7AA RIDS/IEAVTEXS#R

  SYMPTOM              DESCRIPTION
  -------              -----------
  PIDS/5752SC1CM       PROGRAM ID: 5752SC1CM
  RIDS/IEANUC01#L      LOAD MODULE NAME: IEANUC01
  RIDS/IEAVTEXS        CSECT NAME: IEAVTEXS
  AB/S0071             SYSTEM ABEND CODE: 0071
  PRCS/00000010        ABEND REASON CODE: 00000010
  REGS/0E68E           REGISTER/PSW DIFFERENCE FOR R0E: 68E
  REGS/0C7AA           REGISTER/PSW DIFFERENCE FOR R0C: 7AA
  RIDS/IEAVTEXS#R      RECOVERY ROUTINE CSECT NAME: IEAVTEXS
```

Look for a SOFTWARE RECORD for abend X'071' with the following reason codes:

– Reason code X'10': Recovery termination manager (RTM) received control because of an excessive spin. RTM wrote a logrec error record, but did not end any programs and did not request a dump.

– Reason code X'20' or X'30': Because the spin persisted, the system ends the currently running program, might write a logrec error record, and might request a dump.

In a logrec record for an abend X'071', look in the SEARCH ARGUMENT ABSTRACT for the symptom RIDS/IEAVTEXS. If found, look under the heading VARIABLE RECORDING AREA (SDWAVRA) for the following problem data:

- EX SPIN RECORD in the EBCDIC text
- An array of up to 16 pointers: addresses of the functional recovery routines (FRR) on the interrupted FRR stack
- A binary number: index into the FRR stack of the current FRR
- Array of processors causing the spin from SVTSPCP
- Spin loop timeout interval
- Spin loop recovery actions
- Control registers

Example: VERBEXIT LOGDATA Output –– Variable Recording Area

In the following output, the SDWAVRA indicates an excessive spin record and contains an array of pointers to the FRR stack.

```
VARIABLE RECORDING AREA (SDWAVRA)

+000    KEY: 39     LENGTH: 0E
+002    C5E740E2    D7C9D540    D9C5C3D6    D9C4        |EX SPIN RECORD  |

+010    KEY: 37     LENGTH: 04
+012    C6D9D9E2                                        |FRRS            |

+016    KEY: 38     LENGTH: 40
+018    811F43F8    811D74ED    81319571    81086D05    |A..8A...A.N.A._.|
+028    811D74ED    00000000    00000000    00000000    |A...............|
+038    00000000    00000000    00000000    00000000    |................|
+048    00000000    00000000    00000000    00000000    |................|
```

To diagnose the problem, identify the modules involved in the spin, as described in “Step 4: Enabled?” or “Step 5: Analyze Spin Logrec” on page 4-5.

Reference

See *z/OS MVS IPCS Commands* for the VERBEXIT LOGDATA subcommand.

Step 4: Enabled?

Analyze any dump for the type of loop.

Format the dump with a STATUS CPU subcommand. Under the heading CPU(X'nn') STATUS, look for the following:

`DISABLED FOR cccccccc`

System processing was disabled for one or more types of interrupts for the module running at the time of the dump. The system can be disabled for program event recording (PER), I/O, external interrupts (EXT), and machine checks (MCH).

The type is:

- **Disabled loop**: If the system was disabled for I/O or EXT or both.
- **Enabled loop**: If the system was not disabled for I/O or EXT.

Example: STATUS CPU Subcommand Output

In this example, the statement DISABLED FOR PER indicates an ENABLED loop.

```
CPU STATUS:
Warnings regarding STRUCTURE(ASCB) at ASID(X'0001') 00FD5F00:
Located via STRUCTURE(ASVT) at ASID(X'0001') 00F336D0
   Storage not in dump

 PSW=070C2000  8AC2CB2A  (RUNNING IN PRIMARY, KEY 0,  AMODE 31, DAT ON)
     DISABLED FOR PER
   ASID(X'0001') 0AC2CB2A. AREA(PRIVATEX)+02CB2A IN EXTENDED PRIVATE
  ASCB76 at F52080, JOB(CATALOG), for the home ASID
```

Reference

See *z/OS MVS IPCS Commands* for the STATUS subcommand.

Step 5: Analyze Spin Logrec

Analyze a logrec error record for a disabled loop.

Use the excessive-spin logrec error record to identify the module running on the processor causing the spin condition, as follows:

- Locate the 16 FRR address from the stack that was current when the target processor was restarted. These addresses appear after the identification text EX SPIN RECORD at the start of the VRA.
- Identify the current FRR on the stack from the INDEX=*x* value that follows the sixteen addresses. The value of *x* can be 0 through 16.

  If *x* is 0, the stack contains no current FRRs. Otherwise *x* is an index indicating which of the 16 addresses points to the current FRR. For example, if *x* is 2, the second address points to the current FRR.
- Use a storage map to identify the component that owns the FRR at this address.

Example: VERBEXIT LOGDATA Output –– FRR Stack

In the following output, the current FRR is 81319571, which is the third FRR in the stack. This is the current FRR because INDEX=03.

```
+012    C6D9D9E2                                       |FRRS            |

+016    KEY: 38     LENGTH: 40
+018    811F43F8    811D74ED    81319571    81086D05    |A..8A...A.N.A._.|
+028    811D74ED    00000000    00000000    00000000    |A...............|
+038    00000000    00000000    00000000    00000000    |................|
+048    00000000    00000000    00000000    00000000    |................|

+058    KEY: 37     LENGTH: 06
+05A    C9D5C4C5    E77E                               |INDEX=          |

+060    KEY: 38     LENGTH: 01
+062    03                                             |.               |
```

Step 6: Multiprocessor Related?

Analyze a stand-alone dump for a disabled loop.

Loop Analysis

Format the dump with a CPUTRACE subcommand to see the instruction address trace. The output gives the addresses of the instructions that were running and their modules. Check the instructions for a repeating pattern.

If the operator did not request an instruction address trace, continue diagnosis using the system trace table as described in the next step.

Look for loops in the following modules. These disabled loops are used in a multiprocessing (MP) system for waiting and usually do not indicate problems unless they persist. These loops are not valid in a uniprocessor (UP) system and indicate an error.

- **IEAVERI**
- **IEAVESGP**
- **IEAVELK**
- **IEAVEINT**

Loops in other modules indicate problems. Continue diagnosis with step “Step 9: Identify Module and Component” on page 4-8.

Example: SYSTRACE Output

In the following output, notice that processor 04 continues to dispatch SRBs and then WAIT. When processor 05 times out, processor 04 detects an excessive spin condition and issues a restart for processor 05. Processor 05 is in a disabled loop, recognizable by the PSW beginning with X'04'. The address X'13206AA' is the address of the looping module.

```
04 0001 00000000   WAIT
04 0007 00000000   SRB       070C0000 8003B5E8  00000000 00F3D7D0 0018B048
                                                00
04 0001 00000000   WAIT
04 0007 00000000   SRB       070C0000 8003B5E8  00000000 00F3D7D0 0018B048
                                                00
04 0001 00000000   WAIT
05 ******** TIME-GAP OF OVER 0000000A SECS. PREVIOUS TIMESTAMP= A742ED9C
05 0001 00000000  *RST       040C2000 813206AA  01A53D40 20060001 0001229D
                                                20000000 00000004
05 0001 00000000  *RCVY SABN                    04071000 00000030 20000000

05 0001 00000000  *RCVY FRR  040C0000 81319570  04071000 00000030 20000000
```

Reference

See *z/OS MVS IPCS Commands* for the IPCS subcommands.

Step 7: Analyze Dump

Analyze the dump for an enabled loop.

Enabled loops are often quite large and may include several distinct operations, such as I/O, SVCs, module linkages. Because the loop is enabled, it is interrupted, preempted, and resumed many times. This makes the loop pattern difficult to recognize.

Do the following to look for an enabled loop:

- Format the dump with a SYSTRACE subcommand to see the system trace table. Look for a repeating pattern of events in one address space (ASID column). Look for SVC entries with the same PSW address or dispatch entries (DSP, SRB, or SSRB) with PSW addresses that are close together. If found, look at the TCB-ADDR column:

– An address means the event is in a task.

– Zero means the event is in a service request routine.

For SVC entries, use the SVC number to determine the input request for the entry.

Example: SYSTRACE Output

In the following output, the CLKC entries indicate an enabled loop, and because column three is all zeroes, this loop is in SRB mode.

```
01 003E 00000000   CLKC      070C0000 8100765C  00001004 00000000

01 003E 00000000   CLKC      070C2000 81005638  00001004 00000000

01 003E 00000000   CLKC      070C0000 810056E6  00001004 00000000

01 003E 00000000   CLKC      070C0000 80FF0768  00001004 00000000

01 003E 00000000   CLKC      070C0000 80FE4E34  00001004 00000000

01 003E 00000000   CLKC      070C1000 81004BB8  00001004 00000000
```

- Format the dump with an ANALYZE subcommand. Look for many tasks (TCB) waiting for resources held by one job. That job might have a problem, maybe a loop, that keeps it from releasing resources. Note the TCB address.
- Format the dump with the subcommand:

  ```
  SUMMARY KEYFIELD CURRENT
  ```

  Look for an abnormally large amount of request blocks (PRB and SVRB) chained from a task control block (TCB). These blocks suggest a loop that includes several levels of supervisor linkage. Note the TCB address.
- Format the dump with the subcommand:

  ```
  CBSTAT tcb-addr STRUCTURE(TCB)
  ```

  In the TCBs, look for the address of an extended error descriptor block (EED) in the TCBRTM12 field. The recovery termination manager (RTM) uses the EED briefly to pass information from RTM1 to RTM2, then deletes it. Presence of an EED is unusual and could indicate that recovery is in a retry-abend loop.

References

- See *z/OS MVS IPCS Commands* for the IPCS subcommands.
- See *z/OS MVS Diagnosis: Tools and Service Aids* for the system trace entries.
- See *z/OS MVS Diagnosis: Reference* for the SVC summary.
- See *z/OS MVS Data Areas, Vol 2 (DCCB-ITZYRETC)* for the format and contents of the EED.
- See *z/OS MVS Data Areas, Vol 5 (SSAG-XTLST)* for the format and contents of the TCB.

Step 8: Loop Identified?

If dump analysis does not show a loop, continue diagnosis with the procedure in “Chapter 3. Diagnosing a System Hang” on page 3-1.

Step 9: Identify Module and Component

Identify the modules containing each instruction in the loop, as follows:

- *For automatic spin loop recovery, if the dump is an SVC dump*: Use a STATUS WORKSHEET subcommand to obtain the dump title. Obtain the component name, component identifier, or module name from the title.

Example: STATUS WORKSHEET Output

In the following output, the dump title is for the XES component.

```
                             MVS Diagnostic Worksheet
Dump Title: COMPON=IXL,COMPID=5752SCIXL,ISSUER=IXLM1REC,MODULE=IXLM2SP
             ,ABEND=S0071,REASON=00000030

CPU Model 9021 Version A6 Serial no. 300359 Address 03
Date: 03/30/93     Time: 10:32:38  Local

Original dump dataset: SYS1.DUMP32
```

- *For a disabled loop*, use the instruction addresses in the CPUTRACE output.
- *For an enabled loop*:
  - Obtain the PSW addresses from the system trace entries involved in the loop.
  - For ANALYZE output and SUMMARY KEYFIELD CURRENT output, use the TCB address.

  If using a PSW address, ignore the leftmost bit of the leftmost digit. The leftmost bit of the leftmost digit denotes addressing mode and is not part of the address.

Then do one of the following, for each address in the loop:

- If analyzing the dump interactively, use the address in a WHERE subcommand to obtain the name of the load module.
- If analyzing printed output, find the address:
  - In dump output from the LIST or VERBEXIT SUMDUMP subcommand. Look for the CSECT name eye-catcher. IBM module names are generally followed by an assembly date and a product identifier or PTF level, such as HBB4410 or UY01234; most eye-catchers are at the beginning of a module, but some are at the end.
  - In a module listed for the LPAMAP or VERBEXIT NUCMAP subcommand. LPAMAP will list load modules. Use AMBLIST to obtain the offsets of CSECTS within those load modules. NUCMAP lists CSECTs with offsets, but can only be used for modules within the nucleus.

STATUS CPU Output and WHERE Subcommand

In the following output, the PSW identifies the address as X'13206AA'.

```
CPU STATUS:

 PSW=040C2000  813206AA  (RUNNING IN PRIMARY, KEY 0,  AMODE 31, DAT ON)
     DISABLED FOR PER I/O EXT
   ASID(X'0006') 013206AA. IEANUC01.IXLM2SP+07AA IN READ ONLY NUCLEUS
  ASCB1 at FD1780, JOB(*MASTER*), for the home ASID
  ASXB1 at FD1A30 for the home ASID. No block is dispatched
  HOME ASID: 0001 PRIMARY ASID: 0006 SECONDARY ASID: 0006
```

Using the WHERE subcommand, the load module name is IXLM2SP plus an offset of 7AA.

```
ASID(X'0006') 013206AA. IEANUC01.IXLM2SP+07AA IN READ ONLY NUCLEUS
```

Reference

See *z/OS MVS IPCS Commands* for the IPCS subcommands.

Step 10: MVS Component?

From the module prefix, identify the component, subsystem, or product, if provided by IBM.

Use the module name to query the SMP/E zone for a module entry with that module name. If the search does not find a match, the module is not an IBM module. If the search indicates a match, use the FMID to positively identify the product.

Reference

See *z/OS MVS Diagnosis: Reference* to find the IBM component, subsystem, or product for a program identifier or module prefix.

Step 11: Continue Diagnosis

Continue diagnosis as follows:

- If all the addresses are in components of z/OS, continue with “Step 12: Known Problem?” on page 4-10.
- If all the addresses are in an IBM subsystem or product, continue diagnosis with the diagnosis book for the subsystem or product.
- If all the addresses are in components of z/OS and in an IBM subsystem or product, continue with “Step 12: Known Problem?” on page 4-10 and with the diagnosis book for the subsystem or product.
- If any of the addresses are in an installation-provided program, including an installation exit routine, continue diagnosis with that program, using the dump.

  If some addresses are in the program or routine and some in system modules, the loop is probably in the program or routine and includes one or more requests for system services.

Reference

Loop Analysis

See *z/OS MVS Diagnosis: Tools and Service Aids* for information about analyzing an SVC dump for a problem in an installation-provided program.

Step 12: Known Problem?

Search a problem reporting data base to determine if the problem was previously reported.

Use the primary symptom string in VERBEXIT SYMPTOM output, if available, or use the following symptoms:
- Program identifier: PIDS/ccccccccc
- CSECT name or module name: RIDS/cccccccc
- Symptoms created from information in the STATUS CPU output

Note: You can also use free-format search arguments. For more information, see "Searching Problem Reporting Databases" on page 1-4.

Select the problem type on the search tool panel, based on STATUS CPU output:

Problem Type	**STATUS CPU Output**
Disabled loop	DISABLED FOR I/O EXT
Disabled loop	DISABLED FOR I/O
Disabled loop	DISABLED FOR EXT
Enabled loop	DISABLED FOR ccc (not I/O or EXT)

If the search finds no match, remove some symptoms or add some symptoms. Search again. Continue searching for matches by adding and removing symptoms.

If the search finds that the problem was previously reported, request the problem fix. If not, continue with the next step. Use problem data from following steps to create more symptoms; use these symptoms in later searches.

References
- See "Searching Problem Reporting Databases" on page 1-4 for more information.
- See *z/OS MVS IPCS Commands* for the IPCS subcommands.

Step 13: Gather Additional Data, Using IPCS

Collect dump data.

Use IPCS to look at the dump. Use IPCS subcommands in the order indicated by the following list. If using IPCS interactively for an SVC dump requested by an SDUMP or SDUMPX macro, respond yes to the IPCS message that asks if summary data can be used by dump access.

```
STATUS FAILDATA
STATUS SYSTEM
STATUS CPU REGISTERS DATA CONTENTION
STATUS WORKSHEET
ANALYZE
CPUTRACE
SUMMARY TCBERROR
SUMMARY FORMAT
LPAMAP
VERBEXIT NUCMAP
VERBEXIT LOGDATA
VERBEXIT SUMDUMP
SYSTRACE
VERBEXIT MTRACE
Subcommands selected from the following list
VERBEXIT SYMPTOM
```

Before the VERBEXIT SYMPTOM subcommand, add IPCS subcommands, if available, to format data for the looping component. Pick the subcommands from the following list.

Looping Component	**IPCS Subcommand**
Allocation/unallocation	`VERBEXIT ALCMWAIT`
Asynchronous operations manager (AOM)	`VERBEXIT AOMDATA 'TRCDUMP'`

Auxiliary storage manager (ASM)	`ASMCHECK`
	`VERBEXIT ASMDATA`

Availability management	`VERBEXIT AVMDATA`

Callable service requests	`CBFORMAT addr STRUCTURE(CSRCPOOL)`
	`CBSTAT addr STRUCTURE(CSRCPOOL)`

Communications task	`COMCHECK`

Cross-system coupling facility (XCF)	`COUPLE SUMMARY ALL`
	`COUPLE DETAIL ALL`
	`COUPLE EXCEPTION ALL`

Data-in-virtual	`DIVDATA SUMMARY ALL`
Data lookaside facility of VLF	`DLFDATA`
Global resource serialization	`VERBEXIT GRSTRACE`
Input/output supervisor (IOS)	`IOSCHECK`
JES2	`VERBEXIT JES2`
JES3	`VERBEXIT JES3`
MVS message service (MMS)	`VERBEXIT MMSDATA`

z/OS UNIX System Services (OMVS)	`OMVSDATA SUMMARY`
	`OMVSDATA DETAIL`
	`OMVSDATA EXCEPTION`

Real storage manager (RSM)	`RSMDATA SUMMARY`
	`RSMDATA EXCEPTION`

Storage management subsystem (SMS)	`VERBEXIT SMSDATA`
System resources manager (SRM)	`VERBEXIT SRMDATA`

Time sharing option (TSO) `VERBEXIT TSODATA`

Virtual storage management (VSM)
`VERBEXIT VSMDATA`

Virtual lookaside facility (VLF)
`VLFDATA`

Reference

See *z/OS MVS IPCS Commands* for the IPCS subcommands.

Step 14: Gather Additional Data, Trace Analysis

Analyze the trace to determine the range of the loop.

- In an instruction address trace in the CPUTRACE output, find the highest and lowest addresses in the loop.
- In a system trace in SYSTRACE output, identify all entries for the same address space (ASID column) with the same address in the TCB-ADDR column or with zero in TCB-ADDR. For these entries, find the highest and lowest PSW addresses.

Calculate the difference between the highest and lowest addresses. Also, calculate the maximum and minimum offsets from the start of the module for each module involved in the loop.

Step 15: Gather Additional Data, Preceding Events

Analyze the problem data to determine events preceding the loop.

Look in VERBEXIT LOGDATA output and at logrec records for previous problems in the looping component. Look for a SOFTWARE RECORD for a previous abend and a SYMPTOM RECORD for an earlier problem that did not cause an abend. The earlier abend or problem might have caused the problem being diagnosed. Look also at hardware records, such as I/O records, for possible problems.

References

- See *z/OS MVS IPCS Commands* for the VERBEXIT LOGDATA subcommand.
- See *EREP User's Guide* for formatting logrec records.
- See *z/OS MVS Diagnosis: Tools and Service Aids* for information about interpreting logrec error records.

Step 16: Gather Additional Data, Messages

Collect and analyze messages about the problem.

For automatic spin loop recovery, use time stamps to select messages that occurred during the loop. Also, look for messages indicating events leading to the loop. Look in the following:

- The job log
- A TSO/E user's ISPF transaction log or session manager log
- The system log (SYSLOG) for the console with master authority or the alternate console
- VERBEXIT MTRACE dump output, which shows the buffer for system messages

For other loops, use time stamps from the time the problem was reported or the symptoms were noticed.

References

- See *z/OS MVS IPCS Commands* for the VERBEXIT MTRACE subcommand.
- For explanations of the messages:
  - *z/OS MVS System Messages, Vol 1 (ABA-AOM)*
  - *z/OS MVS System Messages, Vol 2 (ARC-ASA)*
  - *z/OS MVS System Messages, Vol 3 (ASB-BPX)*
  - *z/OS MVS System Messages, Vol 4 (CBD-DMO)*
  - *z/OS MVS System Messages, Vol 5 (EDG-GFS)*
  - *z/OS MVS System Messages, Vol 6 (GOS-IEA)*
  - *z/OS MVS System Messages, Vol 7 (IEB-IEE)*
  - *z/OS MVS System Messages, Vol 8 (IEF-IGD)*
  - *z/OS MVS System Messages, Vol 9 (IGF-IWM)*
  - *z/OS MVS System Messages, Vol 10 (IXC-IZP)*
  - *z/OS MVS Dump Output Messages*

Step 17: Gather Additional Data, Contact IBM

Report the problem to IBM, if assistance is needed or if the problem is new.

Provide the following problem data:

- Problem type: disabled loop or enabled loop
- Search argument
- Dump, formatted by IPCS, online or printed
- Range of the loop
- Accompanying messages: identifiers and texts
- Hardcopy log, beginning 30 to 60 minutes before the problem, and master trace
- All printed output and output data sets related to the problem
- Data on any related problems
- Module name(s) and level(s)
- Name and level of the operating system(s) with a list of program temporary fixes (PTF) applied at the time of the problem and all installation modifications, exits, and products with other than Class A service

References

- See “Reporting a Problem to IBM” on page 1-13 for more information.
- See *EREP User’s Guide* for logrec formatting.

Chapter 5. Diagnosing an Output Problem

Symptoms

Your output is incorrect, incomplete, or missing, but messages indicate successful processing.

- **Incorrect output**: The processing produced all the expected output, but some output is incorrect. For example:
  - Some values in a report are wrong
  - The text in a message is incorrect
  - The return or reason code received is not valid
  - The records in are not equal to the records out.
- **Incomplete output**: The processing did not produce all the expected output. For example, a column is missing from a report.
- **Missing output**: Some or all of the expected output is missing. For example, a report is missing.

What You'll Need

- IPCS installed
- Job log
- Input and output data sets for the program
- The JCL for the program
- System log
- Logrec data set

About Using the Procedure

Most output problems occur during installation and testing of new functions or applications.

Output Problem Analysis

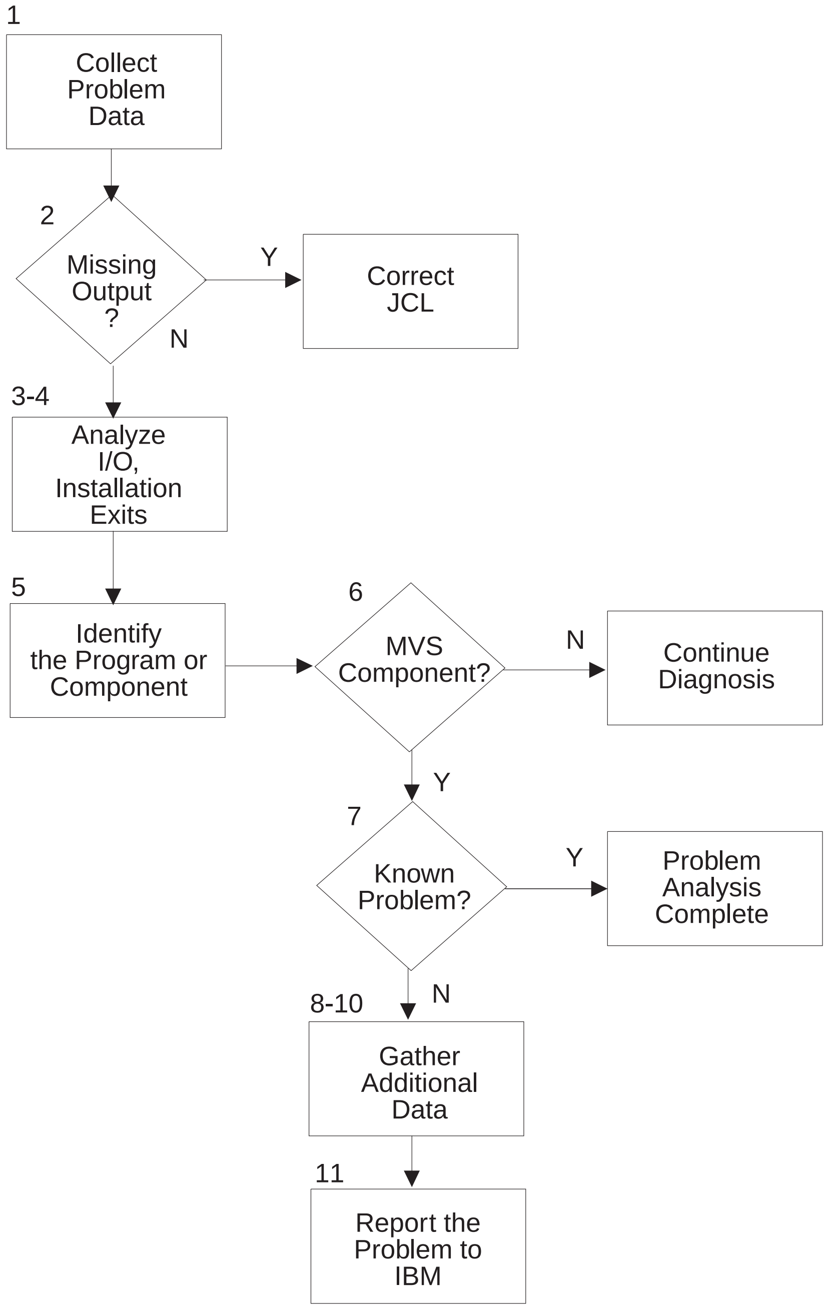

Figure 5-1. Diagnosing an Output Problem

Step 1: Collect Problem Data

Collect all the input and output for the problem.

If the output is a data set, collect the following:

- All input data sets for the program
- Input macros, commands, and statements used to request output from the program
- All output data sets produced by the program

- The job log

If the output is a message, return code, or reason code, collect the incorrect or incomplete message or code.

Step 2: Missing Output?

For missing data set output, analyze data set allocation.

Look for messages in the job log indicating that all data sets used by the program were properly allocated and unallocated. Look for:

- The data set that should have contained the output
- The data set that contained the input used by the program to create the output
- Indications that the output was sent to another data set or different system

If problems are found, correct the JCL. Otherwise, continue diagnosis with the next step in this procedure.

References

- For explanations of the messages, see:
  - *z/OS MVS System Messages, Vol 1 (ABA-AOM)*
  - *z/OS MVS System Messages, Vol 2 (ARC-ASA)*
  - *z/OS MVS System Messages, Vol 3 (ASB-BPX)*
  - *z/OS MVS System Messages, Vol 4 (CBD-DMO)*
  - *z/OS MVS System Messages, Vol 5 (EDG-GFS)*
  - *z/OS MVS System Messages, Vol 6 (GOS-IEA)*
  - *z/OS MVS System Messages, Vol 7 (IEB-IEE)*
  - *z/OS MVS System Messages, Vol 8 (IEF-IGD)*
  - *z/OS MVS System Messages, Vol 9 (IGF-IWM)*
  - *z/OS MVS System Messages, Vol 10 (IXC-IZP)*
- See *z/OS MVS JCL Reference* for JCL coding.

Step 3: Analyze I/O

For incorrect or incomplete data set output, analyze the inputs and outputs.

Do the following:

- Compare the input and the output. For example, if a device model number is wrong in an IOS report, compare it to the model number specified on the in the IODF.
- Compare the output received to examples of the output shown in the user's guide for the request.
- Check the call, command, macro, or statement used to request the output. Make sure that all fields contain desired values.

For missing data set output, check the macro, command, or statement used to request the output. Make sure that the missing output was supposed to be received.

For an incorrect or incomplete message, match the message received to the message in the book that explains it.

Output Problem Analysis

For incorrect return and reason codes, match the codes to the expected codes. Make sure that the code received and the code in the book are both hexadecimal or both decimal.

References

- For output examples, the user's guide for the failing request
- For the message text or a return or reason code in a message:
  - *z/OS MVS System Messages, Vol 1 (ABA-AOM)*
  - *z/OS MVS System Messages, Vol 2 (ARC-ASA)*
  - *z/OS MVS System Messages, Vol 3 (ASB-BPX)*
  - *z/OS MVS System Messages, Vol 4 (CBD-DMO)*
  - *z/OS MVS System Messages, Vol 5 (EDG-GFS)*
  - *z/OS MVS System Messages, Vol 6 (GOS-IEA)*
  - *z/OS MVS System Messages, Vol 7 (IEB-IEE)*
  - *z/OS MVS System Messages, Vol 8 (IEF-IGD)*
  - *z/OS MVS System Messages, Vol 9 (IGF-IWM)*
  - *z/OS MVS System Messages, Vol 10 (IXC-IZP)*
  - *z/OS MVS Dump Output Messages*
  - The message book for a subsystem or program
- For a return or reason code with an abend code, *z/OS MVS System Codes*
- For a return or reason code for a macro:
  - *z/OS MVS Programming: Authorized Assembler Services Reference ALE-DYN*
  - *z/OS MVS Programming: Authorized Assembler Services Reference ENF-IXG*
  - *z/OS MVS Programming: Authorized Assembler Services Reference LLA-SDU*
  - *z/OS MVS Programming: Authorized Assembler Services Reference SET-WTO*
  - *z/OS MVS Programming: Authorized Assembler Services Guide*
  - *z/OS MVS Programming: Assembler Services Reference ABE-HSP*
  - *z/OS MVS Programming: Assembler Services Reference IAR-XCT*
  - *z/OS MVS Programming: Assembler Services Guide*

Step 4: Analyze Installation Exits

Analyze all installation exit routines used in obtaining the output.

Do the following:

- Check for problems in the logic of each routine.
- If no logic problems are found, remove the options that cause each exit routine to be invoked. Rerun the program. If this action stops the problem, the problem is in the exit routine that was eliminated. Continue diagnosis with that routine.

Reference

See *z/OS MVS Installation Exits* for more information on coding installation exit routines.

Step 5: Identify the Program or Component

Identify the program or component involved with the output.

Identify the program or component involved with the output from one of the following:

- For output from a batch job, obtain the program name from the PGM parameter on the JCL EXEC statement.

Example: Obtaining Program Name From the JCL EXEC Statement

In the following example, the name of a program involved with the output is IKJEFT01. It can be found on the PGM parameter of the EXEC statement:

```
//*
//*=====================================================
//* Batch TSO job (PGM=IKJEFT01)
//*=====================================================
//IKJEFT01 EXEC PGM=IKJEFT01,REGION=4096K,DYNAMNBR=50
//*
```

- For output from interactive work, use the command being processed to identify the program.
- For an error message, use the message prefix to identify the program or component or look for the component listed in the message explanation. Look at the preface of any of the z/OS MVS system messages books to find the element or component that the message is associated with and the book where you can find the explanation of the message:

Example: Using the Message Prefix to Identify the Component

If you have an incorrect message with message number AHL002A, you can tell from the preface of *z/OS MVS System Messages, Vol 1 (ABA-AOM)* that the message was issued by GTF, and the explanation can be found in the same book:

Prefix	Component	Book Title	Order Number
AHL	Generalized trace facility (GTF)	*z/OS MVS System Messages, Vol 1 (ABA-AOM)*, *z/OS MVS Dump Output Messages*	SA22-7631 SA22-7590

- For a return or reason code accompanying an abend, see the component listed in the explanation of the code.

Example: Identify the Component Issuing an Abend

00D

> Explanation: An error occurred during processing of a CTRACE or CTRACEWR macro. Register 15 contains xxnnnnxx where nnnn is a reason code that further describes the error.

Reason Code (hex)	**Explanation**
0001	For the CTRACE macro, the parameter list version number is incorrect.
0002	For the CTRACE macro, the component name either does not begin with an alphabetic or national character, or it contains one or more characters that are not alphanumeric or national characters.
Source:	Component trace

Output Problem Analysis

References

- See *z/OS MVS JCL Reference* for the EXEC statement.
- For message prefixes for IBM components, subsystems, and products, see the following books:
  - *z/OS MVS System Messages, Vol 1 (ABA-AOM)*
  - *z/OS MVS System Messages, Vol 2 (ARC-ASA)*
  - *z/OS MVS System Messages, Vol 3 (ASB-BPX)*
  - *z/OS MVS System Messages, Vol 4 (CBD-DMO)*
  - *z/OS MVS System Messages, Vol 5 (EDG-GFS)*
  - *z/OS MVS System Messages, Vol 6 (GOS-IEA)*
  - *z/OS MVS System Messages, Vol 7 (IEB-IEE)*
  - *z/OS MVS System Messages, Vol 8 (IEF-IGD)*
  - *z/OS MVS System Messages, Vol 9 (IGF-IWM)*
  - *z/OS MVS System Messages, Vol 10 (IXC-IZP)*
- See *z/OS MVS System Codes* for explanations of the abend codes.
- See *z/OS MVS Diagnosis: Reference* to relate an IBM component, subsystem or product to a program identifier or module prefix.

Step 6: MVS Component?

Continue diagnosis, depending on whether Step 5 shows an MVS component problem.

Continue diagnosis as follows:

- For a component of z/OS, continue with the next step.
- For an IBM subsystem or product, continue diagnosis with the subsystem or product.
- For an installation-provided program, including an installation exit routine, continue diagnosis with that program.

Step 7: Known Problem?

Search a problem reporting data base to determine if the problem was previously reported.

Search a problem reporting data base to determine if this is a know problem. Use an argument developed from the following symptoms:

- For data set output:
  - Program identifier: PIDS/ccccccccc
  - Input request (macro, command, or statement): PCSS/cccccccccc
  - Data set name: PCSS/cccccccccc
  - Output data record: RECS/cccccccccc
- For message text:
  - Program identifier: PIDS/ccccccccc
  - Message identifier: MSG/cccccccccccc
  - Variable text in the message: VALU/xcccccccc
- For return or reason code:
  - Program identifier: PIDS/ccccccccc
  - Input request (macro, command, or statement): PCSS/cccccccccc
  - Return or reason code received: PRCS/hhhhhhhh
- Identifiers of accompanying messages: MSG/cccccccccccc
- Texts of accompanying messages: VALU/xcccccccc

Note: You can also use free-format search arguments. For more information, see “Searching Problem Reporting Databases” on page 1-4.

Select the problem type on the search tool panel:
- Incorrect output
- Incomplete output
- Missing output

If the search finds no match, remove some symptoms or change the symptoms. Search again. Continue searching for matches by adding, removing, and changing symptoms.

If the search finds that the problem was previously reported, request the problem fix. If not, continue with the next step. Use problem data from following steps to create more symptoms; use these symptoms in later searches.

Reference

See “Searching Problem Reporting Databases” on page 1-4 for more information on developing a search argument.

Step 8: Gather Additional Data, Messages and Logrec

Collect and analyze messages and logrec records about the problem.

Look at any messages or software, symptom, and hardware records for logrec around the time of the problem. Look in the following:
- The job log
- A TSO/E user’s ISPF transaction log or session manager log
- The system log (SYSLOG) for the console with master authority or the alternate console
- Logrec data set, formatted by EREP

Check for:
- I/O errors that could affect the output
- Operator interactions that could affect the output
- Problems with the access method or function involved: For example, VSAM, BTAM, TCAM, JES, or WTO

References
- See *EREP User’s Guide* for formatting logrec records.
- See *z/OS MVS Diagnosis: Tools and Service Aids* for the logrec records.
- For explanations of the messages, see:
  - *z/OS MVS System Messages, Vol 1 (ABA-AOM)*
  - *z/OS MVS System Messages, Vol 2 (ARC-ASA)*
  - *z/OS MVS System Messages, Vol 3 (ASB-BPX)*
  - *z/OS MVS System Messages, Vol 4 (CBD-DMO)*
  - *z/OS MVS System Messages, Vol 5 (EDG-GFS)*
  - *z/OS MVS System Messages, Vol 6 (GOS-IEA)*
  - *z/OS MVS System Messages, Vol 7 (IEB-IEE)*
  - *z/OS MVS System Messages, Vol 8 (IEF-IGD)*
  - *z/OS MVS System Messages, Vol 9 (IGF-IWM)*
  - *z/OS MVS System Messages, Vol 10 (IXC-IZP)*

- *z/OS MVS Dump Output Messages*
- The message book for a subsystem or program

Step 9: Gather Additional Data, Determine Path

Analyze the path the data should take.

For the request being processed, determine the correct path for the data from input to output.

Determine each program and component involved. Check for messages about problems in these programs and components.

If an installation-provided exit routine receives control during the processing, check the routine.

Look at the environment. Specifically, look for recent hardware and software changes to the system and to any applications. A change in one program can affect others; for example, a change to an application that updates a data base affects all other users of the data base.

If needed, re-create the problem, using a SLIP trap and traces to obtain the data needed to isolate the problem.

References

- See the SLIP chapter in *z/OS MVS System Commands* for information on setting a SLIP trap.
- See *z/OS MVS Diagnosis: Tools and Service Aids* for requesting dumps and traces.
- See *z/OS MVS Installation Exits* for exit routines.

Step 10: Gather Additional Data, Teleprocessing

For teleprocessing output problems, analyze the path of the data.

Determine how the data flows through the programs and components that process it and through the systems and hardware. Use this knowledge to re-create the problem, using traces to checkpoint the data at certain spots along the path. Track the data from a point where it was correct to a point where the data stopped or became incorrect.

Reference

See *z/OS Communications Server: SNA Diagnosis V1 Techniques and Procedures* and *z/OS Communications Server: SNA Diagnosis V2 FFST Dumps and the VIT* to diagnose VTAM problems.

Step 11: Report the Problem to IBM

If assistance is needed or if the problem is new, call the IBM Support Center.

Provide the following problem data:

- Problem type: INCORROUT, that is, incorrect, incomplete, or missing output in a data set, message, return code, or reason code

- Search argument
- All input associated with the problem, including all data sets, commands, macros, and statements
- All output associated with the problem, including data sets, reports, and records
- JCL for all data sets involved
- Source code for any exit routine involved
- Accompanying messages: identifiers and texts
- Hard-copy log, beginning 30 to 60 minutes before the problem, or master trace, if not wrapped between the problem and dump
- Name and level of the operating system(s) with a list of program temporary fixes (PTF) applied at the time of the problem and all installation modifications, exits, and products with other than Class A service

References

- See “Reporting a Problem to IBM” on page 1-13 for more information on reporting a problem.
- See *EREP User’s Guide* for logrec formatting.

Chapter 6. Diagnosing a Performance Problem

Symptoms

The system processes jobs slowly. For example, system processing is slow because a program is using an excessive amount of system resources.

What You'll Need

- SVC dump
- Job log
- TSO/E user's ISPF transaction log or session manager log
- System log
- Logrec data set
- RMF, SMF, or other system monitoring program.

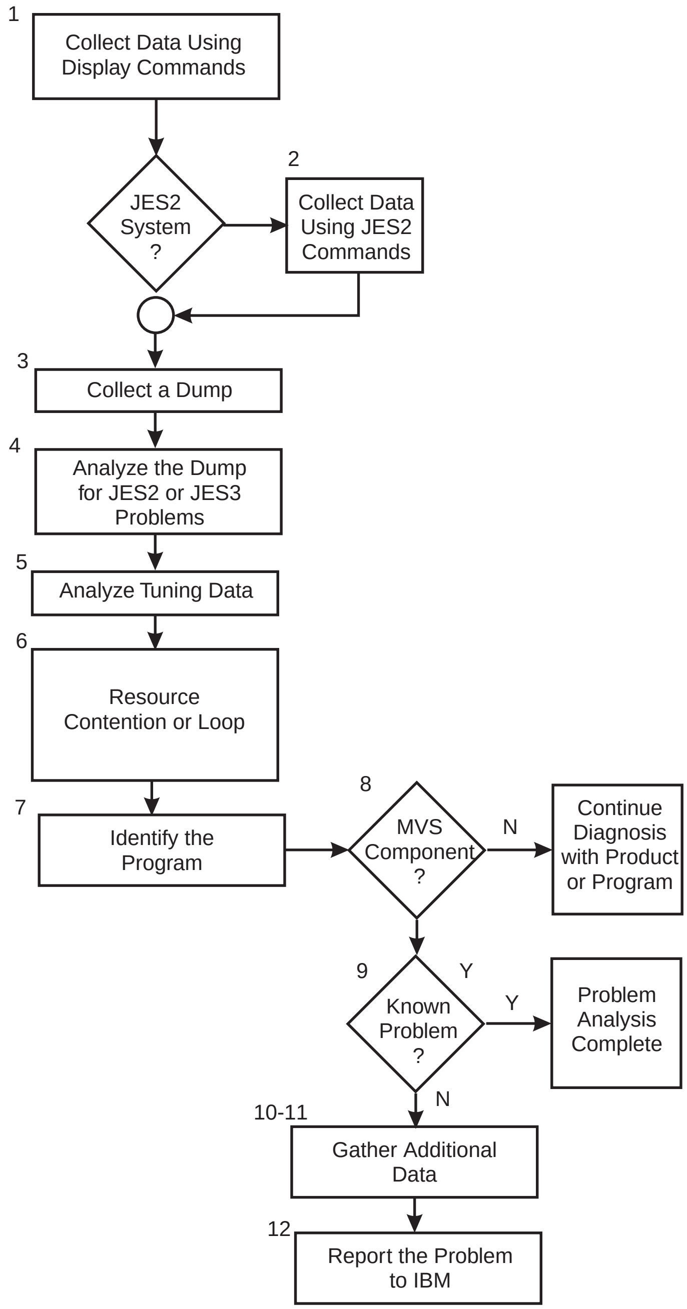
1
Collect Data Using Display Commands
JES2 System ?
2
Collect Data Using JES2 Commands
3
Collect a Dump
4
Analyze the Dump for JES2 or JES3 Problems
5
Analyze Tuning Data
6
Resource Contention or Loop
7
Identify the Program
8
MVS Component ?
N
Continue Diagnosis with Product or Program
9
Y
Known Problem ?
Y
Problem Analysis Complete
N
10-11
Gather Additional Data
12
Report the Problem to IBM

Figure 6-1. Diagnosing a Performance Problem

Step 1: Collect Data Using Display Commands

Ask the operator to collect problem data with DISPLAY commands.

Ask the operator to collect the output from the following DISPLAY commands:

`DISPLAY R,LIST`

Displays outstanding messages requiring operator action. Determine if the system is waiting for an operator action, for example, mounting of a volume. Other jobs may have to wait until the action is completed. In this case, the operator should perform the action.

Example: DISPLAY R,LIST Output

In the following output, there is one message, IEF434D, requiring immediate operator action. The system waits until a valid reply is entered. The operator should enter a valid reply to this message.

```
IEE112I 13.39.37 PENDING REQUESTS    FRAME LAST   F      E   SYS=SY1
RM=0    IM=0     CEM=2     EM=0     RU=0    IR=0    AMRF
ID:R/K     T JOB ID   MESSAGE TEXT
         3 C          *ILR005E PLPA PAGE DATA SET FULL, OVERFLOWING TO
                       COMMON DATA SET
         5 R          *IEF434D CRITJOB - INVALID REPLY. REPLY 'HOLD'
                       OR ' OHOLD'
```

`DISPLAY A,LIST`

Displays detailed information for active jobs and started tasks. A job step name of STARTING indicates that the system has not yet successfully completed initiation of the first step. If unsuccessful initiation of the job continues, diagnose this problem.

Look for the generalized trace facility (GTF) in the started tasks. GTF may slow performance.

Example: DISPLAY A,LIST Output

In the following output, all of the steps have been successfully initiated. However, the performance problem could be due to GTF being active:

```
IEE114I 14.51.49 93.181 ACTIVITY     FRAME LAST   F      E   SYS=SY1
 JOBS     M/S    TS USERS    SYSAS    INITS   ACTIVE/MAX VTAM     OAS
00000    00003    00000      00016    00000    00000/00000       00000
 LLA      LLA      LLA      NSW  S  VLF      VLF      VLF      NSW  S
 JES2     JES2     IEFPROC  NSW  S
 GTF      GTF      IEFPROC  NSW  S
```

`DISPLAY M`

Displays the hardware configuration. Look for the loss of a hardware component indicated by a message on the hard-copy log. This loss may be causing jobs to wait. In this case, correct the hardware problem.

`DISPLAY SLIP`
`DISPLAY SLIP=`*`xxxx`*

Displays the name and status of current SLIP traps. First, enter the DISPLAY SLIP command for a summary of SLIPs running on the system.

Example: SLIPs Displayed Using DISPLAY SLIP Command

```
IEE735I 13.42.00 SLIP DISPLAY        FRAME LAST   F      E   SYS=SY1
ID   STATE    ID   STATE    ID   STATE    ID   STATE    ID   STATE
X013 ENABLED  X028 ENABLED  X0E7 ENABLED  X0F3 ENABLED  X13E ENABLED
X222 ENABLED  X322 ENABLED  X33E ENABLED  X622 ENABLED  X804 ENABLED
X806 ENABLED  X80A ENABLED  X9FB ENABLED  XB37 ENABLED  XD37 ENABLED
XE37 ENABLED
```

Then, pick SLIP IDs that are enabled and enter the DISPLAY SLIP=*xxxx* command for each one to check for enabled PER traps. In this case, disable the traps.

Example: Output of the DISPLAY SLIP=xxxx

In the output from DISPLAY SLIP above, all the SLIPs appeared enabled. The DISPLAY SLIP=X013 command yields the following enabled PER trap with an action of STRACE, which can slow performance:

```
IEE735I 09.14.03 SLIP DISPLAY        FRAME LAST   F      E   SYS=SY1
ID=X013,PER-SB,ENABLED(ACTIVE),ACTION=STRACE,SET BY TSO KLOGAN
```

References

- See *z/OS MVS System Commands* for the SLIP and DISPLAY operator commands.
- See *z/OS MVS Diagnosis: Tools and Service Aids* for information about GTF.

Step 2: Collect Data Using JES2 Commands

In a JES2 system, use JES2 commands to determine why JES2 is not able to schedule work.

Ask the operator to collect problem data with JES2 commands. Use the job entry subsystem display commands to find the status of jobs, queues, printer setups, requirements of SYSOUT data sets, and other problem data.

`$D J1-9999`

Displays the status of jobs. If the display shows that a range of jobs has been held, use the JES2 $A J command to release specific jobs. Or, use the JES2 $A A command to release all jobs in the system.

Example: $D J1-99 Output

In the following output, all of the displayed jobs are being held:

```
$D J1-99

JOB00005  $HASP608 IEBGENER AWAITING HARDCOPY         PRIO HELD 15 ANY
JOB00006  $HASP608 XEQN1    AWAITING XMITTER POK      PRIO HELD 9 ANY
JOB00007  $HASP608 IEBGENER AWAITING HARDCOPY         PRIO HELD 15 ANY
JOB00008  $HASP608 IEBGENER AWAITING EXECUTION A      PRIO HELD 9 ANY
```

`$D S1-9999`

Displays the status of tasks started under JES2:

Example: $D S1-99 Output

The following output shows tasks started under JES2:

```
$D S1-99
 STC00001  $HASP608 SYSLOG   EXECUTING $               PRIO 15 IBM2
 STC00002  $HASP608 $MASCOMM AWAITING HARDCOPY         PRIO 15 ANY
 STC00003  $HASP608 INIT     EXECUTING $               PRIO 15 IBM2
  INITASID=0015
 STC00004  $HASP608 IRRDPTAB ON PRT1                   PRIO  1 IBM2
```

`$D T1-9999`

Displays the status of time-sharing users.

`$D F`
`$D U,PRTS`

Display the status of data set groups queued for output and the status of JES2-controlled local printers. If these displays show that no printers are set up with the needed forms, use the JES2 $T PRTnnnn command to change a printer's setup to the needs of the output forms queue.

Example: $D F and $D U,PRTS Output

In the following output, there is only one item queued for the printers. All of the printers, however, are either drained or halted, which means that none of the printers are started. No printing can occur.

```
$D F

$HASP621 OUT R=LOCAL             F=STD      C=**** T=**** W= (NONE)
$HASP621 PRMODE=LINE     CLASS A=3

$D U, PRTS

$HASP603 PRT1      UNIT=0017,STATUS=HALTED,(STC00004 IRRDPTAB)
$HASP603 PRT2      UNIT=0002,STATUS=DRAINED
$HASP603 PRT3      UNIT=0017,STATUS=DRAINED
$HASP603 PRT4      UNIT=000E,STATUS=DRAINED
$HASP603 PRT5      UNIT=000F,STATUS=DRAINED
$HASP603 PRT6      UNIT=000F,STATUS=DRAINED
```

`$D Q`

Displays the number of queued jobs. If the problem is that jobs are not running because they are held, you can use the JES2 $A command with appropriate parameters to release the jobs.

Reference

See *z/OS JES2 Commands* for the JES2 commands.

Step 3: Collect a Dump

Have the operator request a dump.

If operator action did not correct the problem, have the operator enter a DUMP command to request an SVC dump and reply with SDATA options to dump global resource serialization control blocks and the nucleus. If the problem appears to be in JES2 or JES3, the operator should specify the JES address space in the reply.

Example: Requesting an SVC dump

The following example shows a request for an SVC dump with a sample of the parameters you might use:

```
DUMP COMM=(text)

REPLY id,ASID=1,SDATA=(GRSQ,NUC,CSA,SQA,TRT),END
```

If the operator did not obtain a dump, re-create the problem and request the dump.

References

- See *z/OS MVS Diagnosis: Tools and Service Aids* for requesting an SVC dump.
- See *z/OS MVS System Commands* for the DUMP operator command.

Step 4: Analyze the Dump for JES2 or JES3 Problems

Use IPCS to format the dump for JES2 or JES3 problems.

Performance Problem Analysis

If your dump contains the JES address space and your problem appears to be in JES2 or JES3, format the dump using IPCS as follows:

- For a JES2 system, select JES2 in the Component Analysis panel. Then select JES2 Control Blocks from the JES2 Component Data Analysis panel. From the JES2 Control Block List panel, select the control blocks you wish to format.
- For a JES3 system, select JES3 in the Component Analysis panel. Then select JES3 Control Block Information from the IPCS JES3 - Primary Options panel. You can also use JMF to analyze performance problems for JES3.

References

- See *z/OS JES2 Diagnosis* for information on diagnosing JES2 problems and using IPCS for JES2 diagnosis.
- See *z/OS JES3 Diagnosis* for information on diagnosing JES3 problems, using IPCS for JES3 diagnosis, and JMF.

Step 5: Analyze Tuning Data

Use the display command to see the SRM parameters in the IEAIPSxx, IEAICSxx, and IEAOPTxx parmlib members.

Ask the operator to enter the following command to display the parmlib member names containing current tuning data:

```
DISPLAY DMN
```

Print the SRM parameters in the IEAIPSxx, IEAICSxx, and IEAOPTxx members and analyze their effect on performance. If the SRM parameters are slowing performance, change them.

Example: DISPLAY DMN Output

The following output shows the active IEAIPSxx, IEAICSxx, and IEAOPTxx parmlib members:

```
IEE796I 13.42.55 DOMAIN DISPLAY      FRAME LAST   F      E   SYS=SY1
CURRENT IPS=IEAIPS42,OPT=IEAOPT42,ICS=IEAICS42
DMN  MIN MAX MPLI MPLO CMPL RUA INC NSW OUTU   TWSR     CIDX
  0  999 999  999  999    6   6   0   6    0      3    99.99
  1    2  50   14   15   13  13   0  13    0    300    99.99
  2    3  50    3    3    0   0   0   0    0      0     0.00
  3    1  50    1    1    0   0   0   0    0      0     0.00
```

Use this command output on your system to judge whether the SRM values defined are appropriate for your installation's environment and performance goals.

References

- See *z/OS MVS System Commands* for the DISPLAY operator command.
- See *z/OS MVS Initialization and Tuning Reference* for the parmlib members, SRM parameters, and setting SRM parameters to control performance.

Step 6: Resource Contention or Loop?

Analyze output from RMF or another system monitoring program and look for resource contention and loops.

Use output from RMF, SMF or another system monitoring program to look for problems. Find someone in your installation who is familiar with the program and can interpret the output. The following lists some of the kinds of problems you should look for:

- A program using a lot of storage, whether it is real, virtual, auxiliary or extended storage.
- Data set contention
- ENQ contention
- Tuning problems
- System running over capacity

Perform “Step 6: Check for Resource Contention” on page 3-9 through “Step 11: Check Major System Address Spaces for Dispatchability” on page 3-13 from the procedure on diagnosing a hang or wait.

References

- See *z/OS MVS System Management Facilities (SMF)* for the SMF reports.
- See *RMF User’s Guide* for information.

Step 7: Identify the Program

Identify the program that caused the performance problem.

- For a problem caused by resource contention:
  - Use the ANALYZE output to identify the problem causing the contention:

Example: ANALYZE Output

In the following output, resource #0002, which is device 687, shows an intercept condition and is not running. There is one unit of work waiting for this device.

```
                CONTENTION EXCEPTION REPORT

 JOBNAME=IOS.      ASID=0001  UCBTAPE=00FC72C0

 JOBNAME=IOS.     HOLDS THE FOLLOWING RESOURCE(S):

    RESOURCE #0002:  There are 0001 units of work waiting
                          for this resource
      NAME=I/O Device  687 (TAPE    ) VOLSER=......
      DATA=(IOS) Active I/O with ASSIGN held.
           (IOS) Device not ready.
           (IOS) Intercept condition.

 STATUS FOR THIS UNIT OF WORK:
   IRA10102I This address space is on the SRM IN queue.
```

  - Use other dump output from the steps for a hang or wait.
- For a problem caused by a batch program and identified through JES2 commands, obtain the program name from the PGM parameter on the JCL EXEC statement.

Example: Obtaining the Program Name

In the following example, the name of the program involved with the output is IKJEFT01. It can be found on the PGM parameter of the EXEC statement:

```
//*
//*===================================================
//* Batch TSO job (PGM=IKJEFT01)
//*===================================================
//IKJEFT01 EXEC PGM=IKJEFT01,REGION=4096K,DYNAMNBR=50
//*
```

References

- See *z/OS MVS IPCS Commands* for the ANALYZE subcommand.
- See *z/OS MVS JCL Reference* for the EXEC statement.
- See *z/OS MVS Diagnosis: Reference* to find the IBM component, subsystem, or product for a program identifier or module prefix.

Step 8: MVS/SP Component?

Continue diagnosis depending on what program caused the problem.

Continue diagnosis as follows:

- For a component of z/OS, continue with the next step.
- For an IBM subsystem or product, continue diagnosis with the subsystem or product.
- For an installation-provided program, including an installation exit routine, continue diagnosis with that program.

Step 9: Known Problem?

Search a problem reporting data base for the problem.

Search a problem reporting data base to determine if the problem was previously reported. Use the primary symptom string in VERBEXIT SYMPTOM output, if available, or use the following symptoms:

- Program identifier: PIDS/ccccccccc
- Input request (macro, command, or statement), if one is associated with the problem: PCSS/cccccccccc
- Symptoms created from information in STATUS CPU output

Note: You can also use free-format search arguments. For more information, see “Searching Problem Reporting Databases” on page 1-4.

Select the problem type on the search tool panel:

- Performance

Example: Search Argument Abstract

The following VERBEXIT SYMPTOM output contains the search argument you would use to create your search argument for search a problem reporting data base:

```
                    * * * *  S Y M P T O M  * * * *


Primary Symptom String:

  RIDS/NUCLEUS#L RIDS/IGC018 PIDS/####28422 AB/S0A00 RIDS/IGCT0018#R
  REGS/FE000 PRCS/00000002 VALU/CPAM#BLDL

  Symptom            Symptom data       Explanation
  ---------------    -------------      -----------
  RIDS/NUCLEUS#L     NUCLEUS#L          Routine identifier
  RIDS/IGC018        IGC018             Routine identifier
  PIDS/####28422     ####28422          Component identifier
  AB/S0A00           0A00               ABEND code - system
  RIDS/IGCT0018#R    IGCT0018#R         Routine identifier
  REGS/FE000         FE000              Program register
  PRCS/00000002      00000002           Return code
  VALU/CPAM#BLDL     PAM#BLDL           Error related character value
```

If the search finds no match, remove some symptoms or change the symptoms. Search again. Continue searching for matches by adding, removing, and changing symptoms.

If the search finds that the problem was previously reported, request the problem fix. If not, continue with the next step. Use problem data from following steps to create more symptoms; use these symptoms in later searches.

References

- See “Searching Problem Reporting Databases” on page 1-4.
- See *z/OS MVS IPCS Commands* for the STATUS and VERBEXIT SYMPTOM subcommands.

Step 10: Gather Additional Data, Messages and Logrec

Collect and analyze messages and logrec records.

Collect and analyze messages and logrec records about the problem. Use time stamps to select messages and software, symptom, and hardware records for logrec. Look at any messages or records that occurred before and during the time the problem was first reported. Look in the following:

- The job log
- A TSO/E user’s ISPF transaction log or session manager log
- The system log (SYSLOG) for the console with master authority or the alternate console
- VERBEXIT MTRACE dump output, which shows the buffer for system messages
- VERBEXIT LOGDATA dump output, which formats the logrec buffer
- Logrec data set, formatted by EREP

References

- See *z/OS MVS IPCS Commands* for the VERBEXIT LOGDATA and VERBEXIT MTRACE subcommands.

- See *EREP User's Guide* for formatting logrec records.
- See *z/OS MVS Diagnosis: Tools and Service Aids* for information about logrec.
- For explanations of the messages, see:
  - *z/OS MVS System Messages, Vol 1 (ABA-AOM)*
  - *z/OS MVS System Messages, Vol 2 (ARC-ASA)*
  - *z/OS MVS System Messages, Vol 3 (ASB-BPX)*
  - *z/OS MVS System Messages, Vol 4 (CBD-DMO)*
  - *z/OS MVS System Messages, Vol 5 (EDG-GFS)*
  - *z/OS MVS System Messages, Vol 6 (GOS-IEA)*
  - *z/OS MVS System Messages, Vol 7 (IEB-IEE)*
  - *z/OS MVS System Messages, Vol 8 (IEF-IGD)*
  - *z/OS MVS System Messages, Vol 9 (IGF-IWM)*
  - *z/OS MVS System Messages, Vol 10 (IXC-IZP)*
- See *z/OS MVS Dump Output Messages* for dump output messages.
- See the message book for a subsystem or program.

Step 11: Gather Additional Data, Using IPCS

Collect dump data.

Use IPCS to look at the dump. Use IPCS subcommands in the order indicated by the following list. If using IPCS interactively for an SVC dump requested by an SDUMP or SDUMPX macro, respond yes to the IPCS message that asks if summary data can be used by dump access.

```
STATUS FAILDATA
STATUS SYSTEM
STATUS CPU REGISTERS DATA CONTENTION
STATUS WORKSHEET
ANALYZE
SUMMARY TCBERROR
SUMMARY FORMAT
VERBEXIT LOGDATA
VERBEXIT SUMDUMP
SYSTRACE
VERBEXIT MTRACE
ASMCHECK
CBFORMAT addr STRUCTURE(CSRCPOOL)   If a callable service
CBSTAT addr STRUCTURE(CSRCPOOL)     request was involved
COMCHECK
COUPLE SUMMARY ALL
COUPLE DETAIL ALL
COUPLE EXCEPTION ALL
DIVDATA SUMMARY ALL
DLFDATA
IOSCHECK
OMVSDATA SUMMARY
OMVSDATA DETAIL
OMVSDATA EXCEPTION
RSMDATA SUMMARY
RSMDATA EXCEPTION
VERBEXIT ALCMWAIT
VERBEXIT AOMDATA
VERBEXIT ASMDATA
VERBEXIT AVMDATA
VERBEXIT GRSTRACE
VERBEXIT JES2 or VERBEXIT JES3
VERBEXIT MMSDATA
VERBEXIT SMSDATA
VERBEXIT SRMDATA
VERBEXIT TSODATA
VERBEXIT VSMDATA
VERBEXIT VTAMMAP
VERBEXIT TCAMMAP
VLFDATA
VERBEXIT SYMPTOM
```

Reference

See *z/OS MVS IPCS Commands* for the subcommands.

Step 12: Report the Problem to IBM

If assistance is needed or if the problem is new, call the IBM Support Center.

Provide the following problem data:

- Problem type: performance
- Search argument
- Dump, formatted by IPCS, online or printed
- System responses to DISPLAY and JES commands
- Parmlib members analyzed
- SMF records, if obtained in steps from the hang or wait procedure
- Hard-copy log, beginning 30 to 60 minutes before the problem, or master trace, if not wrapped between the problem and dump
- Logrec records, beginning 30 to 60 minutes before the problem
- All printed output and output data sets related to the problem
- Name and level of the operating system(s) with a list of program temporary fixes (PTF) applied at the time of the problem and all installation modifications, exits, and products with other than Class A service

References

- See “Reporting a Problem to IBM” on page 1-13 for more information.
- See *EREP User’s Guide* for logrec formatting.

Appendix. Notices

This information was developed for products and services offered in the USA.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
USA

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Mail Station P300
2455 South Road
Poughkeepsie, NY 12601-5400
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM’s application programming interfaces.

Programming Interfaces Information

This book documents information that is NOT intended to be used as Programming Interfaces of z/OS.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

- CICS
- DFSMSdfp
- DFSMS/MVS
- Enterprise Systems Architecture/370
- IBM
- IBMLink

- IMS
- IMS/ESA
- MVS/DFP
- MVS/ESA
- MVS/SP
- OS/390
- Processor Resource/Systems Manager
- PRISM
- Resource Link
- RMF
- VTAM
- z/OS

UNIX is a registered trademark of The Open Company in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

Index

Readers' Comments — We'd Like to Hear from You

z/OS
MVS Diagnosis: Procedures

Publication No. GA22-7587-00

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	□	□	□	□	□

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	□	□	□	□	□
Complete	□	□	□	□	□
Easy to find	□	□	□	□	□
Easy to understand	□	□	□	□	□
Well organized	□	□	□	□	□
Applicable to your tasks	□	□	□	□	□

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? □ Yes □ No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Name

Address

Company or Organization

Phone No.

Readers' Comments — We'd Like to Hear from You

GA22-7587-00

Cut or Fold
Along Line

Fold and Tape **Please do not staple** Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
Department 55JA, Mail Station P384
2455 South Road
Poughkeepsie, NY
12601-5400

Fold and Tape **Please do not staple** Fold and Tape

Cut or Fold
Along Line

Program Number: 5694-A01

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.